UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

TRABAJO FIN DE GRADO

Descubrimiento y propagación de información y preferencias en redes sociales y entornos de recomendación

Rocío Cañamares Pérez

Tutor: Pablo Castells Azpilicueta

Mayo 2014

Resumen

Los datos que se utilizan en la ejecución y evaluación de algoritmos de recomendación tienen fuertes sesgos en la distribución de las observaciones. Es relevante, por tanto, entender cómo se generan estos sesgos de cara a ajustar y evaluar dichos algoritmos. El trabajo que se presenta en esta memoria ha consistido en diseñar un escenario, y un conjunto de modelos y herramientas que permiten materializarlo, constituyendo una plataforma de soporte al estudio sistemático de dichas distribuciones y muy en particular la influencia que tienen sobre ellas los fenómenos de propagación de información en redes sociales.

El escenario desarrollado es una generalización de otros modelos propuestos y estudiados por otros autores, frente a los cuales se contrasta y valida nuestra propuesta. Entre las diferencias originales del modelo que desarrollamos, destacamos la inclusión de los gustos de los usuarios, que permiten comparar la distribución resultante de votos observados con las opiniones reales (observadas o no) de dichos usuarios.

La implementación del modelo se ha realizado en Java, utilizando diversas librerías de dominio público. Dicha implementación incluye una simulación que permite la creación y evolución de las distribuciones generadas, así como una interfaz gráfica para su visualización y análisis en tiempo real.

Una vez implementado, y a fin de validar la verosimilitud y potencial explicativo de nuestra propuesta, se ha utilizado el modelo para ajustar distribuciones de conjuntos de datos públicos del campo de los sistemas de recomendación (MovieLens, Twitter, Epinions y Foursquare) con el objetivo de recrear el posible proceso y la combinación de factores que se esconden tras las distribuciones resultantes que se observan en estas colecciones.

Por último, se ha realizado un análisis preliminar de las relaciones y dependencias entre los parámetros de configuración del modelo y las variables de salida resultantes. En particular, se ha visto que la comunicación entre los usuarios influye muy notablemente en cómo se distribuye el descubrimiento de información y, con él, la generación de ratings. También comprobamos que la forma de la red social es clave en la distribución del descubrimiento y en particular en la velocidad a la que se produce.

Palabras clave: recomendación, red social, propagación en redes sociales, modelización probabilística, ajuste de curvas, simulación.

Abstract

The data (observations of user-item interactions) used in the implementation and evaluation of recommendation algorithms come from observations with strongly biased distributions. These biases are mostly ignored in the literature of the area, yet understanding the generation of these biases is important to better interpret the results from empirical experiments, and eventually adjust such algorithms and/or the methodologies by which they are evaluated. The work documented in the present thesis consists in designing a scenario along with a set of models and tools to implement it, for studying such distributions and, more specifically, how they are influenced by the spreading of information through social networks.

This scenario is a generalization of previously proposed and studied models by other authors, against which we relate and validate our proposal. One of the specific features of the proposed model, that distinguishes it from prior ones, is the incorporation of user preferences in the dynamics of the modelled system, by which we aim to study the role of users' tastes in the resulting distributions of observed ratings, and the relation thereof with the real (observed or not) users' opinions.

We use the Java programming language, along with several public domain libraries, to implement the model. The implementation includes a simulation that allows the creation and evolution of the generated distributions, as well as a user interface with extensive dynamic graphical views of the system state and statistics,allowing to visualize and analyze them in real time.

In order to validate the plausibility and explanatory power of our proposal, we have used the model to fit distributions of public real datasets in the recommender systems field (MovieLens, Twitter, Foursquare and Epinions) in order to reproduce possible processes and combinations of factors that may lie behind the distributions observed int this collections.

Finally, we have performed a preliminary analysis of relationships and dependencies between model configuration parameters and output variables. Specifically, we have seen that under preference-dependent user behavior, the communication between users has a great influence on how the information discovery – and consequently, the rating generation – is distributed. Also the shape of the social network appears to be important in the discovery distribution and, in particular, the speed at which it discovery runs through the network.

Keywords: recommendation, social network, propagation in social networks, probabilistic modeling, curve fitting, simulation.

Agradecimientos

En primer lugar quiero dar las gracias al IRG (Information Retrieval Group) por darme la oportunidad de trabajar con ellos y de desarrollar allí este proyecto. Me gustaría destacar la excelente labor de Pablo, mi tutor, que desde el principio me ha guiado y se ha involucrado con mi trabajo, aconsejándome cada día nuevos caminos a tomar, resolviendo mis dudas sin tardanza y corrigiendo mis resultados.

Mis compañeros también han sido clave, quizá no tanto en el TFG, pero desde luego sí a lo largo de la carrera. Así, quiero agradecer a Mario y a Gallego por enseñarme a programar, más concretamente, a Gallego por pasarse hasta las tantas depurando punteros conmigo para programación I y a Mario por contestar siempre mis dudas y preguntas, más o menos amablemente y tras remarcar lo tontas que son (sigue siendo Mario). También quería agradecer su amistad a Lara, Cris y Julia que me han hecho pasar muy buenos momentos. En particular a Lara, mi compañera de prácticas, por todos esos fines de semana, viernes hasta las ocho y Semanas Santas acabando prácticas y redactando memorias.

Por último, quiero agradecer el apoyo de toda mi familia. En particular, quiero destacar la labor de mis padres, su paciencia escuchando mis quejas, su atención y su apoyo. Ambos han sido unos magníficos modelos de superación y de éxito para mí.

A ellos, y al resto de personas que me han ayudado a pasar por estos cinco años de carrera y han participado de una forma u otra en que este proyecto se lleve a cabo les quiero dedicar mis más sinceros agradecimientos.

Rocío

Índice

1. Introducción 1
1.1 Motivación 1
1.2 Objetivos 3
1.3 Estructura del trabajo 4
2. Estado del arte 5
2.1 Redes sociales 5
2.1.1 Representación 5
2.1.2 Topología y métricas 6
2.1.3 Formación 6
2.2 Propagación 9
2.2.1 Propagación de rumores 9
2.2.2 Epidemias 10
2.3 Recomendación 12
2.3.1 Conjuntos de datos públicos de ratings reales 13
2.3.2 Recomendación y relevancia 14
2.4 Modelo de formación de opinión 16
3. Diseño del modelo 17
3.1 Planteamiento general 17
3.2 Definición formal 20
3.2.1 Relevancia 20
3.2.2 Descubrimiento 20
3.2.3 Consumición 22
3.2.4 Votación 22
3.2.5 Comunicación 23
3.2.6 Dependencias entre las variables 23
4. Marco de simulación 25
4.1 Inicialización de la simulación 25
4.1.1 Tipos y construcción de la red social 25
4.1.2 Distribución de relevancia 26
4.2 Bucle central de la simulación 28
4.2.1 Descubrimiento 29
4.2.2 Consumición 30
4.2.3 Votación 30

4.2.4 Comunicación: flujo en la red social. ... 30
4.3 Condiciones de parada de la simulación ... 36
4.4 Valores de salida de la simulación ... 36
4.4.1 Ajuste ... 37
4.5 Tabla de parámetros ... 39
5. Detalles de implementación ... 41
5.1 Módulos ... 41
5.1.1 Proceso central de la simulación ... 41
5.1.2 Grafos ... 41
5.1.3 Recomendadores ... 44
5.1.4 Estadísticos ... 44
5.1.5 Interfaz de usuario ... 44
5.2 Base de datos ... 45
5.3 Interfaz de usuario ... 47
5.3.1 Ventana principal (1-4) ... 47
5.3.2 Ventana de parámetros (6) ... 48
5.3.3 Visualización de la distribución del grado (8) ... 48
5.3.4 Ventana de ajuste (7) ... 48
5.4 Visualización de la propagación en grafo ... 50
5.5 Librerias ... 54
6. Experimentos ... 55
6.1 Reproducción del experimento de Doerr (2012) ... 55
6.2 Reproducción de una epidemia ... 58
6.3 Ajuste de datos reales ... 59
6.3.1 Restricción en el tamaño de los datasets ... 60
6.3.2 MovieLens ... 61
6.3.3 Epinions ... 61
6.3.4 Twitter ... 61
6.3.5 Foursquare ... 62
6.3.6 Resultados ... 63
6.4 Exploración de parámetros ... 65
6.4.1 Relación entre comunicación y votación ... 66
6.4.2 Único amigo vs. todos los amigos ... 67
6.4.3 Nivel de descubrimiento exógeno ... 69
6.4.4 Grafo ... 70
7. Conclusiones ... 73
7.1 Resumen y contribuciones ... 73

7.2 Trabajo futuro 74
7.2.1 Modelo 74
7.2.2 Análisis de resultados 75
Referencias 77

Índice de Figuras

Figura 1. Visualización de la topología y la distribución del grado de un grafo con 100 nodos y grado promedio 4, generado según el modelo Erdös. 7

Figura 2. Visualización de la topología y la distribución del grado de un grafo con 100 nodos y grado promedio 4, generado según el modelo Barabási. 8

Figura 3. Comparativa con la red social Orkut obtenida al reproducir el experimento de Doerr et al. 10

Figura 4. Crecimiento logístico clásico de la curva del modelo IS de epidemias. 11

Figura 5. Distribución de ratings (en escalas lineal y logarítmica) y distribución de frecuencias de MovieLens. 13

Figura 6. Distribución de ratings y de frecuencias de Epinions, Netflix y Last.fm. 15

Figura 7. Esquema del modelo propuesto. 23

Figura 8. Forma de la función power law (clásica y modificada) para distintos valores del parámetro α. 27

Figura 9. Diagrama de clases del módulo de grafos. 42

Figura 10. Diseño de la estructura interna del grafo completo. 43

Figura 11. Diseño de la estructura interna de ArrayGraph. 43

Figura 12. Diagrama ER de la base de datos. 46

Figura 13. Intefaz de la aplicación principal. 49

Figura 14. Intefaz de la visualización cualitativa de la propagación en grafo. 53

Figura 15. Comparativa con la red social Orkut obtenida al reproducir el experimento de Doerr et al. 57

Figura 16. Comparativa con la red social Twitter obtenida al reproducir el experimento de Doerr et al. 57

Figura 17. Ajuste de los conjuntos de datos públicos MovieLens, Epinions, Twitter y Foursquare. 63

Figura 18. Valor de $p(rate)$ en función de $p(tell|R)$ pero manteniendo constante $p(tell)$. 67

Figura 19. Curva de descubrimiento en función de si los usuarios hablan con todos sus amigos o sólo con uno cuando se comunican. 67

Figura 20. Curva de descubrimiento y distribución de relevancia en función de si los usuarios hablan con todos sus amigos o sólo con uno cuando se comunican. 68

Figura 21. Curva de descubrimiento en función del ratio entre películas vistas y descubiertas. 69

Figura 22. Curva de descubrimiento y distribución de relevancia en función del ratio entre películas vistas y descubiertas. 70

Figura 23. Grafo y subgrafos de Facebook. 71

Figura 24. Comparativa de las curvas de descubrimiento en función del tipo de grafo. 72

Índice de Tablas

Tabla 1. Datos de los datasets públicos de MovieLens, Epinions, Netflix y Last.fm. ... 14

Tabla 2. Lista de las variables aleatorias del modelo con sus correspondientes descripciones. 20

Tabla 3. Dependencias entre las distintas variables. 24

Tabla 4. Resumen de los distintos modos de comunicación y los parámetros que los determinan. 35

Tabla 5. Lista de parámetros de la simulación 40

Tabla 6. Datos volumétricos de las redes sociales Orkut y Twitter. 56

Tabla 7. Datos de los conjuntos de datos MovieLens, Epinions, Twitter y Foursquare tras el preprocesamiento realizado para ajustar los tamaños. 63

Tabla 8. Asignación de parámetros cuyo valor es común al ajuste de todos los datasets. 64

Tabla 9. Asignación de parámetros que difieren del ajuste de un dataset a otro. 65

Tabla 10. Variación realizada de los parámetros $p(tell|R)$ y $p(tell|\neg R)$ de forma que el nivel de comunicación $p(tell)$ se mantenga constante. 66

Tabla 11 Características de los grafos Barabási y Facebook, incluyendo los subgrafos de este último. 70

1. Introducción

1.1 Motivación

A lo largo de la última década hemos asistido al surgimiento y desarrollo de un nuevo fenómeno tecnológico y social: las redes sociales online (Scott 2011). Muchas consecuencias se derivan de este fenómeno, pues está incidiendo en la evolución de las formas de comunicarnos y relacionarnos con las personas de nuestro entorno social. Una consecuencia inmediata –a la que siguen otras– de las redes online es que los procesos que ya tenían lugar en redes offline de personas se han hecho observables, y por tanto susceptibles de ser estudiados.

Uno de los fenómenos importantes en los que las redes, tanto digitales como naturales, juegan un papel clave es la propagación de información, la cual ha sido objeto de estudio e interés de muchas disciplinas del conocimiento (sociólogos, economistas, matemáticos, físicos, tecnólogos…). En el caso de las redes online la propagación de información va acompañada por otro hecho relevante: las reacciones de los usuarios a la nueva información que les llega pueden registrarse, dejando tras de sí una *huella digital* (votos, comentarios, opiniones, relaciones con otros usuarios…) que puede analizarse y utilizarse para el desarrollo de nuevas funcionalidades. En particular, este rastro constituye una fuente de información muy valiosa para desarrollar las llamadas tecnologías de recomendación, que dan al usuario una asistencia personalizada proactiva para acceder a información y oportunidades de potencial interés en muy diversos ámbitos.

Los sistemas de recomendación tienen actualmente casi dos décadas y media de desarrollo y se han vuelto familiares para cualquier usuario de aplicaciones tan comunes como Youtube, Spotify, multitud de tiendas online (Amazon, Fnac, etc.), Google News, Twitter, Facebook y LinkedIn, AdSense, Netflix, Smart TV, y un largo etcétera de aplicaciones y tecnologías de uso masivo. En este tiempo se ha formado una comunidad investigadora en este campo que ha enfocado sus esfuerzos a aspectos tales como, primordialmente, el diseño de algoritmos de recomendación efectivos que optimicen el acierto con los gustos de los usuarios, así como el desarrollo de principios y metodologías para evaluar este acierto y otras cualidades deseables de los recomendadores. Los avances algorítmicos documentados en la literatura se vienen contrastando y orientando con estudios empíricos sobre conjuntos de datos públicos o propietarios, centrando el análisis experimental en comparativas minuciosas de las mediciones finales resultantes. Sin embargo apenas ningún trabajo se ha ocupado de analizar cómo se han generado los conjuntos de datos de prueba, qué propiedades estadísticas presentan (más allá del volumen y la densidad de los datos), qué posibles sesgos pueden presentar en su distribución, cómo influyen éstos en el comportamiento de los algoritmos, y en las mediciones que sobre éstos a su vez se realizan para evaluar su rendimiento.

La huella digital tiene, de hecho, fuertes sesgos en la distribución de observaciones (Steck 2010), como se puede observar en las Figuras 5 y 6 de la sección 2.3.1 en las que se muestran las distribuciones de votos de varios conjuntos de datos públicos. La frecuencia de interacción de los usuarios con los productos susceptibles de ser recomendados sigue típicamente una distribución de Pareto en la que unos pocos

productos populares concentran la atención de los usuarios (y por tanto los registros de acceso e interacción), mientras que el resto de productos forman una larga cola de elementos poco o menos conocidos, con mucha menor presencia en los registros de interacción. En consecuencia, la mayoría de los algoritmos de recomendación actualmente conocidos tienden a concentrar sus sugerencias entre los productos populares, y las metodologías de evaluación actuales basadas en métricas como la precisión (Cremonesi 2010) tienden a premiar este comportamiento.

El presente trabajo se plantea la pregunta de cómo se forman los sesgos observables en los datos que los sistemas de recomendación toman como entrada. En este contexto, nuestro trabajo contempla la hipótesis fundamental de que la propagación de información y en particular la comunicación entre personas, juega un papel esencial en las dinámicas que determinan la distribución de la huella digital. Gran parte de la información que nos llega lo hace a través de nuestro entorno social. Puesto que el conocimiento de algo es condición primera para interactuar con ello, nos planteamos aquí estudiar el efecto que las dinámicas de difusión de información en red pueden producir en la generación y distribución de interacciones entre usuarios e ítems recomendables, esto es, el input para los sistemas de recomendación. Son conocidas y estudiadas las dinámicas de red que dan lugar a fenómenos virales de popularidad, y creemos que estas mismas dinámicas pueden explicar las distribuciones desiguales de los datos que alimentan a los algoritmos de recomendación. Esta posible relación entre propagación y huella digital no ha sido prácticamente estudiada hasta ahora. La mayoría de trabajos realizados hasta el momento suelen tomar como punto de partida dicha huella, sin cuestionar su distribución. Únicamente se ha encontrado un trabajo que estudie dicha distribución (Blattner 2012), que analizamos en la Sección 2.4 sobre el estado del arte.

El trabajo tiene por tanto una doble motivación. Por un lado, explicar y describir los sesgos en los datos de interacción es una cuestión relevante para saber cómo tratar con ello en un sistema de recomendación. Por ejemplo, ¿es deseable o indeseable que un algoritmo de recomendación se deje llevar por el sesgo de lo popular? La respuesta depende de cómo se ha llegado a formar esa popularidad, y qué factores han incidido en ella: una película puede hacerse popular por su calidad, por su campaña de promoción, por las dinámicas del boca a boca y las topologías de red por las que discurren, etc. El trabajo que aquí se aborda busca dar elementos de juicio que ayuden a comprender y dilucidar mejor estas cuestiones con vistas a, en un futuro, tenerlas en cuenta en el desarrollo y evaluación de los algoritmos de recomendación.

Por otro lado, el problema plantea un escenario de análisis y modelización de dinámicas de propagación en red con características específicas donde, como veremos, introducimos elementos tales como la valoración subjetiva del usuario hacia la información que pasa por él, y decisiones intermedias de los usuarios tales como prestar o no atención a la información que les llega, o emitir valoraciones positivas o negativas del tipo que puede explotar un recomendador. Nuestro trabajo extiende así estudios desarrollados en esta línea en el campo del análisis de redes sociales y los procesos de difusión, en una dirección que, por estos elementos específicos –a la vez que razonablemente generales–, comporta novedad respecto a trabajos previos.

Cabe destacar que ambos problemas - el estudio de los sesgos en los datos de interacción y el análisis de las dinámicas de propagación en red – podrían abordarse desde, en principio, dos puntos de vista bastante diferentes. Por un lado, se podrían realizar experimentos basados en usuarios reales y observar sus pautas de comportamiento y sus decisiones, llegando incluso a realizar encuestas que informen

acerca de los motivos de esos comportamientos. Este tipo de trabajos suponen un gran coste y una necesidad de recursos que no son siempre accesibles. Alternativamente, el estudio puede ceñirse a un nivel teórico, basándose en hipótesis sobre los comportamientos reales, y reproducirse mediante simulaciones. El trabajo llevado a cabo se ajusta a esta última opción, en línea con muchos otros trabajos que realizan estudios similares (Doerr 2012).

Un último comentario en referencia a la motivación y el planteamiento de este trabajo, es que se ha desarrollado en el contexto de una línea de investigación orientada al desarrollo y evaluación de algoritmos de recomendación y al análisis de preferencias que se lleva a cabo en el grupo de investigación IRG (Information Retrieval Group).

1.2 Objetivos

El presente trabajo tiene por objeto diseñar un escenario de recomendación que permita estudiar los fenómenos de descubrimiento y propagación en redes sociales, así como la generación y distribución de ratings. Se incorpora, también, el concepto de relevancia (gustos de los usuarios), como dato característico de este modelo, con el objetivo de comparar las distribuciones obtenidas con dicha relevancia. Además, se realiza una implementación de dicho escenario que sea ejecutable mediante una simulación y que permita ajustar distribuciones de ratings reales extraídas de conjuntos de datos públicos. También se estudiarán las diferencias y semejanzas con otros trabajos realizados anteriormente y se contrastarán los resultados obtenidos.

Con más detalle, los objetivos de este trabajo son:

- Definir un modelo que describa una dinámica de interacción de personas con otras personas e ítems de un dominio dado, en la que se contemplen los matices y elementos propios que dan lugar a los datos que toma como entrada un sistema de recomendación. El modelo se formulará de tal manera que estén presentes en él los elementos clave de nuestras hipótesis de partida, tales como el comportamiento de los usuarios o la calidad de los objetos (ítems) con los que interactúan.
- Completar el modelo con una dinámica procedimental que permita simular procesos basados en una determinada configuración del modelo, y observar los estados resultantes y su evolución.
- Implementar un marco completo de simulación que permita configurar, instanciar e inicializar el modelo, ejecutar dinámicas, y visualizar diferentes vistas de la evolución del estado del sistema.
- Analizar a través del modelo y los elementos adicionales desarrollados en el trabajo (simulador, visualizaciones, etc.) la relación entre factores fundamentales tales como los gustos de los usuarios, su comportamiento en la comunicación con otros usuarios y la interacción con los ítems del dominio.
- Comparar el modelo con estudios previos relacionados, identificando y comprendiendo con exactitud las diferencias, y comprobando la equivalencia con éstos como caso particular de nuestro modelo.
- Obtener distribuciones de datos semejantes a las observables en conjuntos de datos públicos mediante configuraciones específicas de los parámetros de nuestro modelo. En otras palabras, conseguir ajustes de estas distribuciones con instanciaciones particulares de nuestro modelo, lo que equivale a formular

explicaciones potenciales de las condiciones que pudieran haber dado lugar a las distribuciones observadas

1.3 Estructura del trabajo

El documento se estructura de la siguiente forma.

En el apartado 2 (Estado del arte) se explican los conocimientos necesarios para entender el resto del trabajo así como la situación actual de los temas que serán recurrentes a lo largo del documento: redes sociales, propagación y recomendación.

En el apartado 3 (Diseño del modelo) se diseña el modelo objeto del trabajo y se exponen y motivan las variables que intervendrán en dicho escenario.

En el apartado 4 (Marco de simulación) se incorporan al modelo los patrones de comportamiento y las dinámicas necesarias para poder realizar una simulación del mismo.

En el apartado 5 (Detalles de implementación) se explica la implementación del escenario descrito en los apartados anteriores así como la interfaz y las librerías y demás tecnologías que han sido utilizadas para llevarlo a cabo.

En el apartado 6 (Experimentos) se exponen los experimentos realizados mediante la simulación que incluyen la exploración y el análisis de las relaciones y la influencia de los parámetros, la comparación de los resultados obtenidos con los de otros trabajos anteriores y los ajustes de las distribuciones de conjuntos de datos públicos.

Para concluir, en el apartado 7 (Conclusiones) se resume el trabajo y se sintetizan las conclusiones que de él se extraen. También se introducen los posibles caminos a seguir en un trabajo futuro.

2. Estado del arte

Introducimos y contextualizamos aquí en primer lugar una serie de conceptos que serán recurrentes a lo largo del documento. En concreto, hablaremos de redes sociales, de los fenómenos de propagación que en ellas se producen, introduciremos brevemente el concepto de recomendación y mostraremos algunos conjuntos de datos reales y las distribuciones que presentan. Por último, explicaremos el modelo de formación de opinión propuesto en el artículo de Blattner et al (2012) que comparte nuestro objetivo de simular la distribución de ratings reales, y mencionaremos los aspectos que lo diferencian del escenario que se propone en este trabajo.

2.1 Redes sociales

El estudio y la evolución de las redes sociales no ha seguido un curso lineal, sino que, como otras muchas disciplinas afectadas por la tecnología, se ha visto impulsada de forma exponencial en los últimos tiempos. Aunque las redes sociales son un concepto anterior a cualquier tipo de tecnología moderna y están presentes desde los inicios de la humanidad, el estudio metódico de las mismas es mucho más reciente.

Fueron los sociólogos los primeros en mostrar interés por este campo. Así, desde principios del siglo XX hasta la década de los 90, investigan las interacciones sociales y empiezan a formar el concepto de red social. En la década de 1990, sin embargo, los físicos toman el relevo y, desde un punto de vista más analítico y formal, definen los modelos topológicos, las métricas y demás teorías y formulaciones que impulsan significativamente el campo de las redes sociales. En la década de los 2000, de la mano del desarrollo de internet, las redes sociales dejan de ser únicamente un objeto de interés de pensadores, sociólogos y científicos y pasan a primer plano para la mayoría de las personas. Tal es así que, hoy en día, las redes sociales online son la fuente principal de información, intercambio de ideas y generación de opiniones de muchas personas (Anagnostopoulos 2008). Además, han sido el motor de varios movimientos sociales de gran alcance (Howard 2011).

Todas estas circunstancias hacen que el estudio de las redes sociales, su formación y los fenómenos de propagación que en ellas se producen, haya cobrado gran relevancia.

2.1.1 Representación

En el caso típico, una red social se modeliza como un grafo que puede ser dirigido o no. Esta estructura surge de manera natural dado que tenemos un conjunto de elementos (usuarios) con conexiones entre sí. Así, los nodos del grafo serán los usuarios y las aristas indicarán que dos usuarios se conocen y puede haber comunicación entre ellos.

Aunque las redes sociales más comunes son no dirigidas, pues si un usuario conoce a otro entonces ese otro le conoce a él, podemos encontrar ejemplos de redes sociales dirigidas, como la estructura de seguidores de Twitter, en donde el hecho de que un usuario siga a otro no implica necesariamente que sea seguido por él.

En la mayoría de estudios, y en este trabajo en particular, se consideran grafos no ponderados y simples (una única arista entre dos nodos). Sin embargo, hay casos en los que podría tener sentido considerar aristas con pesos (p.e. frecuencia de iteración) y/o multigrafos (vínculos de diferentes tipos entre las mismas personas. Respecto al sentido de las aristas, se estudiarán tanto grafos dirigidos como no dirigidos.

2.1.2 Topología y métricas

Resulta clave para diferenciar unas redes de otras estudiar la forma o topología que adoptan, para lo cual nos valemos de diversas métricas y estadísticas. Las más simples atienden a aspectos como el tamaño, la densidad o la distribución del grado. Esta última es una de las más utilizadas e indica, para cada grado, el número de nodos que lo tienen. Se ha observado que la mayoría de redes sociales no poseen una distribución de grado uniforme. De hecho, dicha distribución suele tener la forma de una función *power law* o distribución de Pareto, esto es, hay pocos nodos con muchos vecinos (grado alto) y la mayoría de vértices tienen un grado en torno a la media. Este tipo de distribución se puede observar en la Figura 2.b.

Respecto a las métricas más elaboradas, éstas pueden aplicarse tanto a nodos individuales como al grafo globalmente y existen multitud de ellas que atienden a factores muy diversos. Las métricas individuales permiten además un análisis micro de regiones de la red, o de individuos concretos, identificando personas que tienen un determinado papel de importancia bajo diferentes ángulos medidos por diferentes métricas, p.e. personas con potencial de influencia vs. personas con potencial de mediación, etc.

En este trabajo, la métrica básica de interés que vamos a tener en cuenta es el coeficiente de clustering, por ser una de las más representativas y que mejor caracteriza la topología de los grafos.

El coeficiente de clustering local de un nodo u refleja la cohesión del entorno de dicho nodo y se calcula como la probabilidad de que dos vecinos de u elegidos al azar estén conectados.

$$C(u) = \frac{nº\ conexiones\ entre\ vecinos\ de\ u}{nº\ conexiones\ posibles\ entre\ vecinos\ de\ u}$$

El coeficiente de clustering global del grafo se calcula como el promedio del coeficiente de clustering de los nodos:

$$C(G) = avg_u C(u)$$

2.1.3 Formación

El estudio de las redes sociales no atañe únicamente a sus características actuales (métricas), sino que también se centra en el proceso de formación de la red social. Las redes sociales son cualquier cosa menos estáticas, y de la dinámica que da lugar a su formación se derivan propiedades (p.e. topológicas) de la red resultante. Es por ello de interés entender los procesos por los que se desarrollan las estructuras de red, y los estudiosos del campo han dedicado importantes esfuerzos a este punto desde mediados del siglo XX. El estudio de la formación de redes se basa en la consideración de este fenómeno como un proceso estocástico en el que las redes crecen según un modelo probabilístico: se introducen progresivamente personas y conexiones en determinados puntos de la red con arreglo a sucesos probabilísticos en los que intervienen variables aleatorias sujetas a determinados modelos de distribución.

Los modelos de formación de red tienen interés y utilidad desde diferentes puntos de vista. Entre otros, por un lado, tienen un potencial explicativo de la generación de ciertas características de las redes reales. Por otro, son útiles para simular redes con ciertas características con el tamaño y densidad que nos convenga. Permiten además contrastar el comportamiento de redes reales con estos modelos más sencillos, como de hecho haremos en este trabajo.

Son múltiples los diferentes modelos de red que se han postulado y analizado a lo largo de los años. Los dos modelos más omnipresentes por su sencillez y representatividad son el modelo de enlace aleatorio (random attachment) propuesto por Erdös y Rényi (Erdös 1959), y el modelo de enlace preferente (preferential attachment) propuesto por Barabási y Albert (Barabási 1999). En el presente trabajo tomaremos estos dos modelos como referencia comparativa en diferentes puntos de nuestro estudio. Ambos toman como dato el número total de usuarios y el grado promedio del grafo que ha de resultar, y se diferencian en la forma de colocar las aristas.

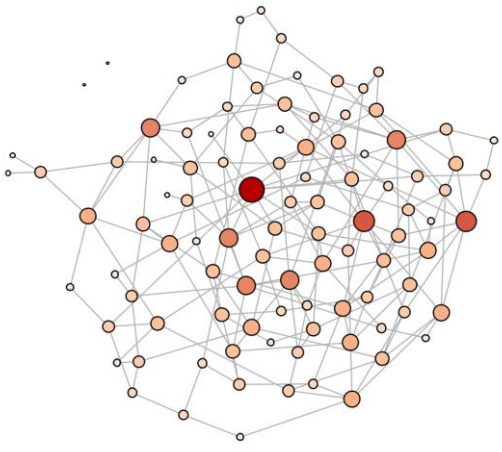
2.1.3.1 Modelo Erdös-Rényi (enlace aleatorio)

La forma más simple de crear un grafo del que conocemos el número de vértices es elegir, según una probabilidad uniforme y de forma iterativa, los pares de nodos entre los que colocar una arista. Una vez que se ha alcanzado el grado promedio preestablecido, dejamos de añadir aristas al grafo.

Este modelo de formación genera un grafo con una distribución del grado binomial, es decir, los grados de los nodos se concentran en torno al grado promedio y son muy raros los nodos con grados muy altos o muy bajos.

En la Figura 1.a podemos observar un grafo de 100 nodos con un grado promedio de 4 generado según el modelo Erdös. El tamaño y el color de cada nodo reflejan el número de vecinos que tiene: cuantos más vecinos, mayor es el radio y más oscuro es el color. En la Figura 1.b se muestra la distribución del grado de dicho grafo que, como vemos, coincide con la forma de una distribución binomial.

a) Grafo Erdos



b) Distribución del grado del grafo Erdos

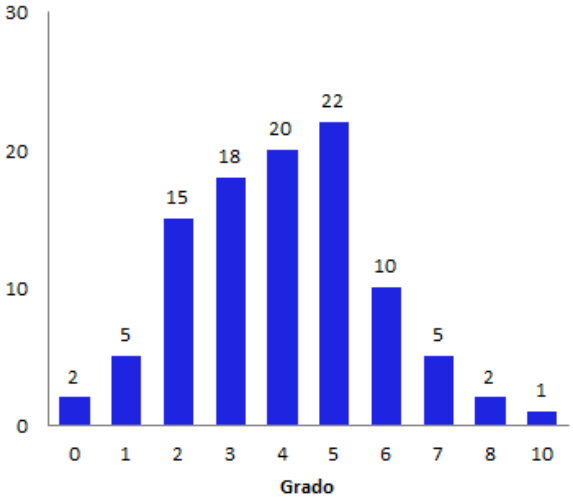


Figura 1. Visualización de la topología y la distribución del grado de un grafo con 100 nodos y grado promedio 4, generado según el modelo Erdös.

Observando ambas figuras notamos que, tal y como habíamos predicho, se detectan diferentes grados pero, en general, las diferencias no son muy amplias, sobre todo si las

comparamos con el grafo generado según el modelo Barabási (Figura 2.a) para un grafo del mismo tamaño y densidad.

2.1.3.2 Modelo Barabási-Albert (enlace preferente)

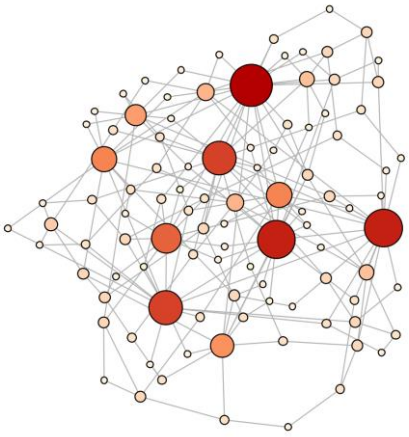
Como ya se ha explicado en la sección 2.1.2 en la mayoría de redes sociales el grado no se distribuye de forma uniforme sino que sigue una distribución de Pareto, por lo que el modelo de formación de Erdös no sirve para generar este tipo de redes.

El modelo Barabási propone la siguiente alternativa. Así como en el modelo de enlace aleatorio todos los nodos del grafo están presentes desde el principio, en el modelo de Barabási los nodos se van añadiendo a la red uno por uno. Cuando añadimos un nodo, lo conectamos con un cierto número fijo de vértices del grafo. La forma de elegir los nodos con los que conectar al nuevo vértice no es uniforme, sino que se eligen con más probabilidad los nodos que más contactos tienen. De esta forma, los nodos con muchos vecinos tienden a acumular todavía más y el grafo resultante alcanza la distribución deseada. De este proceso resulta una distribución del grado que sigue una ley de potencias (power law).

A pesar de que el modelo Barabási aproxima bastante bien la distribución del grado de las redes sociales reales, no ocurre lo mismo con el coeficiente de clustering. Ambos, Erdös y Barabási, presentan un coeficiente de clustering muy inferior al deseado.

En la Figura 2.a observamos un grafo con 100 nodos y un grado promedio de 4 generado siguiendo el modelo Barabási. Nuevamente, el tamaño y color de los nodos informa acerca de su grado. Esta visualización junto con la gráfica de la distribución del grado, que vemos en la Figura 2.b, nos permite observar que la mayor parte de los nodos posee un grado pequeño, de en torno a 2, mientras que hay unos pocos nodos que destacan por tener muchos más vecinos.

a) Grafo Barabási



b) Distribución del grado del grafo Barabási

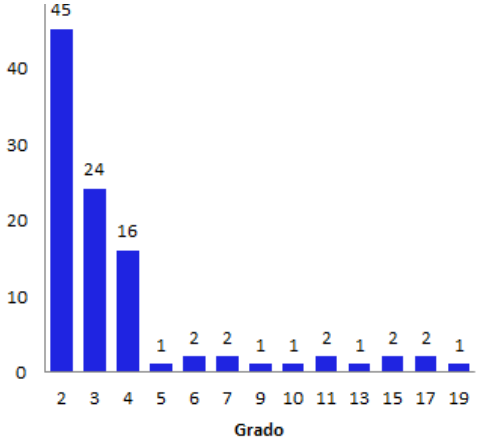


Figura 2. Visualización de la topología y la distribución del grado de un grafo con 100 nodos y grado promedio 4, generado según el modelo Barabási.

Además de los modelos de formación de Erdös y Barabási se han formulado innumerables alternativas en la literatura, tales como como el modelo Eppstein o el modelo Kleinberg. Este último parte de un anillo y añade arcos aleatoriamente con preferencia por distancias cortas, produciendo un coeficiente de clustering mayor que el modelo Barabási. En nuestras pruebas con diferentes modelos aleatorios hemos observado que no se producen diferencias significativas con estos modelos alternativos

respecto a lo que observamos con los modelos Erdös y Barabási, por lo que a la hora de estudiar el comportamiento del modelo ante distintas redes sociales, nos limitaremos a los modelos Erdös y Barabási y a grafos provenientes de redes sociales reales, como Twitter u Orkut.

2.2 Propagación

Uno de los aspectos por los que las redes sociales han obtenido gran relevancia en los últimos tiempos está relacionado con los fenómenos de propagación que en ellas se producen. Aunque existen gran cantidad de objetos susceptibles de ser trasmitidos en una red social (estados, opinión, información, etc.), nosotros vamos a centrarnos en la propagación de información.

A continuación se exponen dos modelos concretos de propagación que están estrechamente relacionados con el escenario desarrollado: un conocido estudio sobre la propagación de rumores a través de redes sociales (Doerr 2012) y el modelo IS de propagación de epidemias (Newman 2010).

2.2.1 Propagación de rumores

Doerr et al (2012) exponen un experimento cuyo objetivo es estudiar la velocidad de propagación de información en las redes sociales. Este trabajo es relevante para nuestro estudio porque el experimento que plantea es un caso particular del modelo desarrollado en este trabajo, motivo por el cual reproduciremos dicho experimento como tal caso particular, con el objetivo de contrastar los resultados. Aunque se explicará más en detalle en la sección 6.1 de experimentos, donde se realiza la comparación entre ambos modelos, comentamos brevemente a continuación en qué consiste dicho experimento.

El estudio de Doerr et al considera una red social y establece un estado inicial en el que un usuario posee una determinada pieza de información. A continuación se inicia un proceso en el que se itera sucesivamente sobre todos los usuarios, y cada uno elige al azar uno de sus vecinos con el que comunicarse e intercambiar la información que posee. El experimento consiste en contrastar, en función del tipo de grafo, el número de usuarios que conocen la información tras cada iteración – una iteración es el recorrido de todos los usuarios. El estudio compara la velocidad de propagación en las redes sociales reales de Orkut y Twitter con otro tipo de redes teóricas: Barabási, Erdös y un grafo completo.

En la Figura 3 se muestran los resultados obtenidos al repetir la parte del experimento de Doerr et al que hace referencia a la comparativa de distintas topologías de red con la de la red social Orkut. En esta gráfica se puede observar, para cada tipo de grafo, el número de usuarios que conocen la información tras cada iteración. En el caso de Orkut, se observa una velocidad superior a la alcanzada en los grafos Erdös y completo, pero que va a la par de la obtenida por el grafo Barabási. La comparativa con la red social Twitter presenta una gráfica muy parecida, que ya veremos en el apartado de experimentos, pero en este caso la velocidad de Twitter es bastante superior al resto de grafos teóricos. Es decir, parece que la propia topología de las redes sociales reales, similar a la obtenida con el modelo de formación Barabási, favorece la rápida propagación de información.

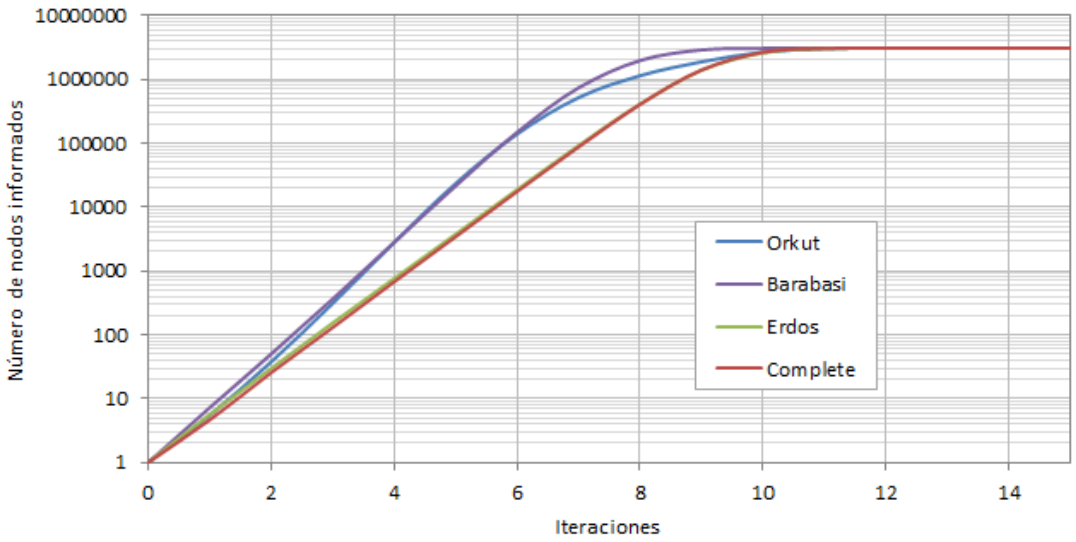


Figura 3. Comparativa con la red social Orkut obtenida al reproducir el experimento de Doerr et al.

Además de este experimento empírico, en el artículo se realiza un análisis teórico y se determina analíticamente la velocidad a la que se propaga la información en un grafo Barabási. Así mismo, se concluye que la gran velocidad alcanzada en este tipo de grafos, en comparación con el resto, se debe a la existencia de nodos puente, esto es, nodos con pocos contactos que unen dos o más nodos con muchos contactos.

Sin embargo, pese a la gran importancia de los resultados analíticos, en nuestro caso nos centraremos en reproducir las curvas obtenidas por el experimento para los distintos tipos de grafos, dejando a un lado el análisis teórico del mismo.

2.2.2 Epidemias

Uno de los fenómenos de propagación en redes sociales más estudiados son las epidemias. De hecho, existen varias formas de modelizar una epidemia, que dependen de las variables y estados que se consideren: infectados, susceptibles, aislados, periodos de incubación, etc. Nosotros vamos a considerar el modelo IS, infectado-susceptible, (Newman 2010) porque, bajo esta modelización, las epidemias son un caso particular del modelo que se presenta en este trabajo y, por tanto, sirven para contrastarlo. De hecho, presentan equivalencia con el experimento de la ACM, explicado anteriormente.

El modelo IS es el más simple de todos los modelos de epidemias y sólo considera dos tipos de individuos: los infectados, que portan la enfermedad, y los susceptibles que, como su propio nombre indica, son susceptibles de adquirir la enfermedad cuando contactan con un infectado. Además, según el modelo IS, el contagio es instantáneo, es decir, cuando un susceptible y un infectado se relacionan, el susceptible se contagia automáticamente. Se descarta, por tanto, la opción de que las personas sanen, mueran o sean inmunes, así como los períodos de incubación. Por otro lado, los contactos se producen al azar entre cualquier par de nodos, es decir, la red social es un grafo completo en el que todos los usuarios están conectados entre sí.

Para deducir la ecuación de una epidemia según el modelo IS tenemos en cuenta los siguientes variables:

- β: Número de contactos por individuo y por unidad de tiempo.

- X: Número de infectados.
- S: Número de susceptibles.
- N: Número total de individuos.

Asumiendo una distribución uniforme, la probabilidad de que un infectado contacte con un susceptible es S/N y, teniendo en cuenta que contacta con β personas por unidad de tiempo, un infectado contagia en promedio a $\beta S/N$ susceptibles. Como hay X infectados, se produce un incremento de $X\beta S/N$ infectados por unidad de tiempo. Esto nos permite plantear la siguiente ecuación diferencial:

$$\frac{dX}{dt} = \frac{X \cdot \beta \cdot S}{N}$$

Al resolverla obtenemos la siguiente ecuación que establece el número de infectados en función del tiempo:

$$X(t) = \frac{x_0 e^{\beta t}}{N - x_0 + x_0 e^{\beta t}}$$

donde x_0 indica el número inicial de infectados.

Generalmente y salvo valores poco comunes de x_0 (0 o negativos), esta ecuación produce lo que se denomina una *curva logística* como la mostrada en la Figura 4, es decir, presenta un crecimiento exponencial al principio, que se corresponde con la fase inicial de la epidemia en la que la mayoría de la población es susceptible, y posteriormente se satura y el crecimiento se ralentiza, pues cada vez es más difícil para la enfermedad encontrar nuevos individuos susceptibles.

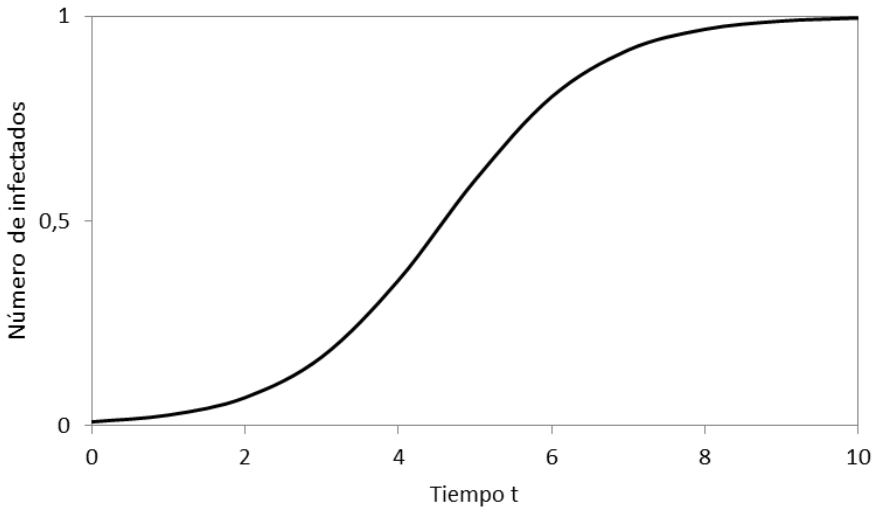


Figura 4. Crecimiento logístico clásico de la curva del modelo IS de epidemias. Indica el número de infectados por unidad de tiempo siendo los parámetros de la epidemia $\beta = 1$, $x_0 = 1$ y $N = 100$.

La diferencia principal entre el modelo de epidemias y el propuesto por Doerr et al es que la red social subyacente se considera un grafo completo en el caso de las epidemias, ya que todos los usuarios pueden contactar con todos, mientras que Doerr et al considera distintos tipos de grafos. En el resto de aspectos son muy similares, pues la enfermedad de la epidemia se puede modelizar como la pieza de información que se

propaga en el modelo de Doerr et al, los infectados como los usuarios que conocen la información y los suceptibles como los usuarios que la desconocen. Cuando un usuario que conoce la información se comunica con otro que no lo hace, le transmite la información, es decir, le contagia.

A modo de avance de las innovaciones propuestas por nuestro modelo frente a estos otros dos modelos que se detallarán en los apartados 6.1 y 6.2 de experimentos, cabe destacar la introducción de los gustos de los usuarios en el proceso de comunicación, que deja de ser inmediato – dos usuarios contactan y seguidamente intercambian información acerca del ítem – a estar modelizado por probabilidades y opiniones – dos usuarios contactan pero la probabilidad de que intercambien información acerca de un cierto ítem depende de lo que opinen de él. Además, el propio modelo de comunicación (usuarios con los que contactar, ítems de los que hablar, momento en que hablar…) es más complejo y presenta muchas más variaciones. Por último, nuestro modelo no es únicamente un modelo de propagación, sino que también simula la generación de interacciones usuario – ítem (opiniones, comentarios, votos…).

2.3 Recomendación

Aunque elaborar algoritmos de recomendación no es un objetivo del trabajo, construir un escenario que permita estudiar cómo se distribuyen los datos que reciben estos algoritmos sí lo es, por lo que introducimos aquí unas breves nociones sobre qué es un recomendador.

Un recomendador es un elemento que *observa* al usuario y su interacción con los ítems y, en función de esas observaciones, determina los posibles ítems que le pueden interesar y se los recomienda. Las observaciones acerca de un usuario (entrada de los recomendadores) pueden ser datos de muy diverso tipo: ítems que visita, tiempo que está observando cada uno de ellos, votos directos (ratings) con un valor numérico o booleano (me gusta/no me gusta), relaciones con el resto de usuarios, encuestas ad hoc, etc.

En función de los datos de entrada del recomendador y su forma de interpretarlos, podemos clasificar los recomendadores en dos grandes tipos:

- Personalizados: tienen en cuenta las características individuales de cada usuario, además de otros factores, para adivinar sus intereses futuros. Dentro de los recomendadores personalizados existen varios subgrupos:
  - Recomendadores de filtrado colaborativo: en base a las opiniones del resto de usuarios y las similitudes que presentan con las valoraciones del usuario, se adivina su opinión acerca de los ítems que no conoce pero que personas con opiniones similares han valorado.
  - Recomendadores basados en contenido: se basan en las características de los ítems valorados por el usuario para estimar su opinión acerca de otros con características similares. No tienen en cuenta las opiniones de otros usuarios.
  - Recomendadores basados en red social: se basan en la opinión de los amigos del usuario en la red social para determinar la opinión del usuario.
  - Recomendadores híbridos: son combinaciones de los anteriores que equilibran sus virtudes y debilidades.

- No personalizados: recomiendan lo mismo a todos los usuarios, sin tener en cuenta sus características individuales. Estas recomendaciones pueden realizarse en base a muchos factores, entre los cuales destacan la opinión de la mayoría, las críticas de los expertos o las prioridades del proveedor que tiene interés en promocionar un cierto producto.

Sin embargo, como ya hemos dicho, los algoritmos de recomendación no son el objetivo de este trabajo, por lo que no se profundizará más de lo hasta ahora explicado. Si el lector tiene interés en el tema puede consultar (Ricci 2011, Adomavicius 2012).

Por nuestra parte, estamos más interesados en los datos que reciben como entrada dichos recomendadores. Estos datos suelen ser, por lo general, ratings numéricos realizados por los usuarios acerca de los ítems. Para simplificar, y sin pérdida de generalidad a efectos de los objetivos de nuestro estudio, asumiremos ratings binarios, que toman un valor 1 si el ítem le gusta al usuario y un valor 0 si no le gusta.

Existen varios conjuntos de datos públicos que comentaremos brevemente a continuación y en los que se pueden observar votaciones reales de usuarios.

2.3.1 Conjuntos de datos públicos de ratings reales

Existen muchos conjuntos de datos públicos en lo que se refiere a ratings realizados por usuarios reales, entre los más conocidos y utilizados en los estudios de recomendación se encuentran MovieLens, Epinions, Netflix y Last.fm.

Como hemos explicado anteriormente, la distribución de observaciones de la interacción usuario – ítem suele presentar fuertes sesgos. Analicemos las distribuciones de estos cuatro datasets públicos para ver a qué nos referimos exactamente:

2.3.1.1 MovieLens

MovieLens[1] es un conjunto de datos públicos de puntuaciones por usuarios a películas. Se trata posiblemente del dataset más utilizado en la literatura y la tradición investigadora en el campo de los sistemas de recomendación. Consta de 4000 usuarios, 6000 películas y 1 millón de ratings.

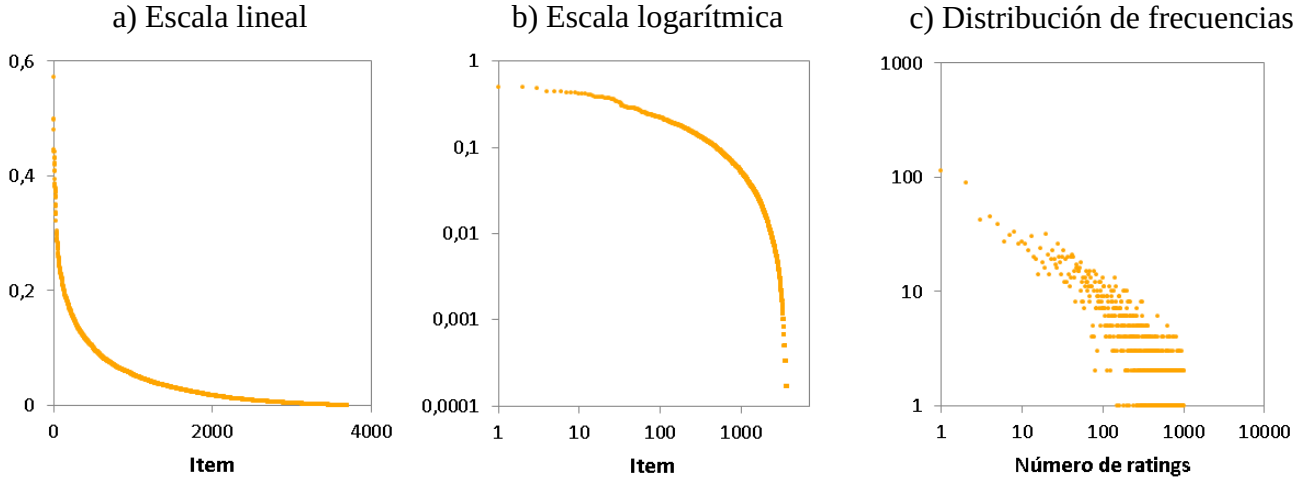


Figura 5. Distribución de ratings (en escalas lineal y logarítmica) y distribución de frecuencias de MovieLens.

[1] http://grouplens.org/datasets/movielens

En las Figuras 5.a y 5.b se muestra la distribución de ratings de MovieLens, que indica, para cada ítem, el ratio de usuarios que lo han votado. Observamos claramente que sigue la forma de una distribución de Pareto (o power law) que ya hemos mencionado varias veces. Así, observamos unos pocos ítems votados por muchos usuarios mientras que la mayoría reciben muy pocas valoraciones. Existe, por tanto, un gran sesgo hacia las películas más votadas, es decir, los usuarios tienen a votar más a las películas que ya tienen muchos votos.

Otra forma de mostrar esta información es mediante la distribución de frecuencias que consiste en indicar para cada posible número de ratings el número de ítems que han recibido dichos ratings. En esta escala es fácil distinguir las distribuciones con forma de power law, pues presentan forma de tendencia lineal, tal y como se puede observar en la Figura 5.c donde se muestra la distribución de frecuencias de MovieLens.

2.3.1.2 Epinions, Netflix y Last.fm

Epinions[2], Netflix[3] y Last.fm[4] son, al igual que MovieLens, conjuntos de datos públicos de votaciones de usuarios. En el caso de Epinions los ítems sobre los que se realizan las votaciones son artículos de todo tipo, en Netflix son películas y series de televisión y en Last.fm son canciones y artistas.

En la Tabla 1 podemos observar las estadísticas de estos tres conjuntos de datos, junto con las de MovieLens.

Dataset	Nº. de usuarios	Nº. de ítems	Nº. de ratings
MovieLens	4.000	6.000	1.000.000
Epinions	132.000	1.560.144	13.668.319
Netflix	148.000	17.000	100.000.000
Last.fm	360.000	160.000	18.000.000

Tabla 1. Datos de los datasets públicos de MovieLens, Epinions, Netflix y Last.fm.

En la Figura 6 observamos que las distribuciones de ratings y de frecuencias de Epinions, Netflix y Last.fm presentan la misma forma de power law que la de MovieLens, es decir, tienen el mismo sesgo hacia las películas más votadas.

2.3.2 Recomendación y relevancia

Observamos que, tras los conceptos de recomendador y rating, se esconde uno mucho más elemental: la relevancia o, lo que es lo mismo, los gustos de los usuarios. Después de todo, son precisamente estos gustos los que los recomendadores persiguen adivinar y los ratings reflejar. El análisis de dichos gustos puede abordarse desde dos puntos de vista:

- Generación: los gustos de los usuarios suelen ser variables y dependen de cuestiones como la publicidad, el propio criterio personal, las críticas, la opinión de los amigos, etc. Además, los gustos de los usuarios se desarrollan y evolucionan cuando interactúan con los ítems, es decir la experiencia altera los gustos de las personas.

[2] http://www.trustlet.org/wiki/Epinions_datasets
[3] file://raptor.ii.uam.es/collections/datasets/Datasets/netflix/others/viewtopic.php.htm
[4] file://raptor.ii.uam.es/collections/datasets/Datasets/Last.fm/360K%20Users/index.html

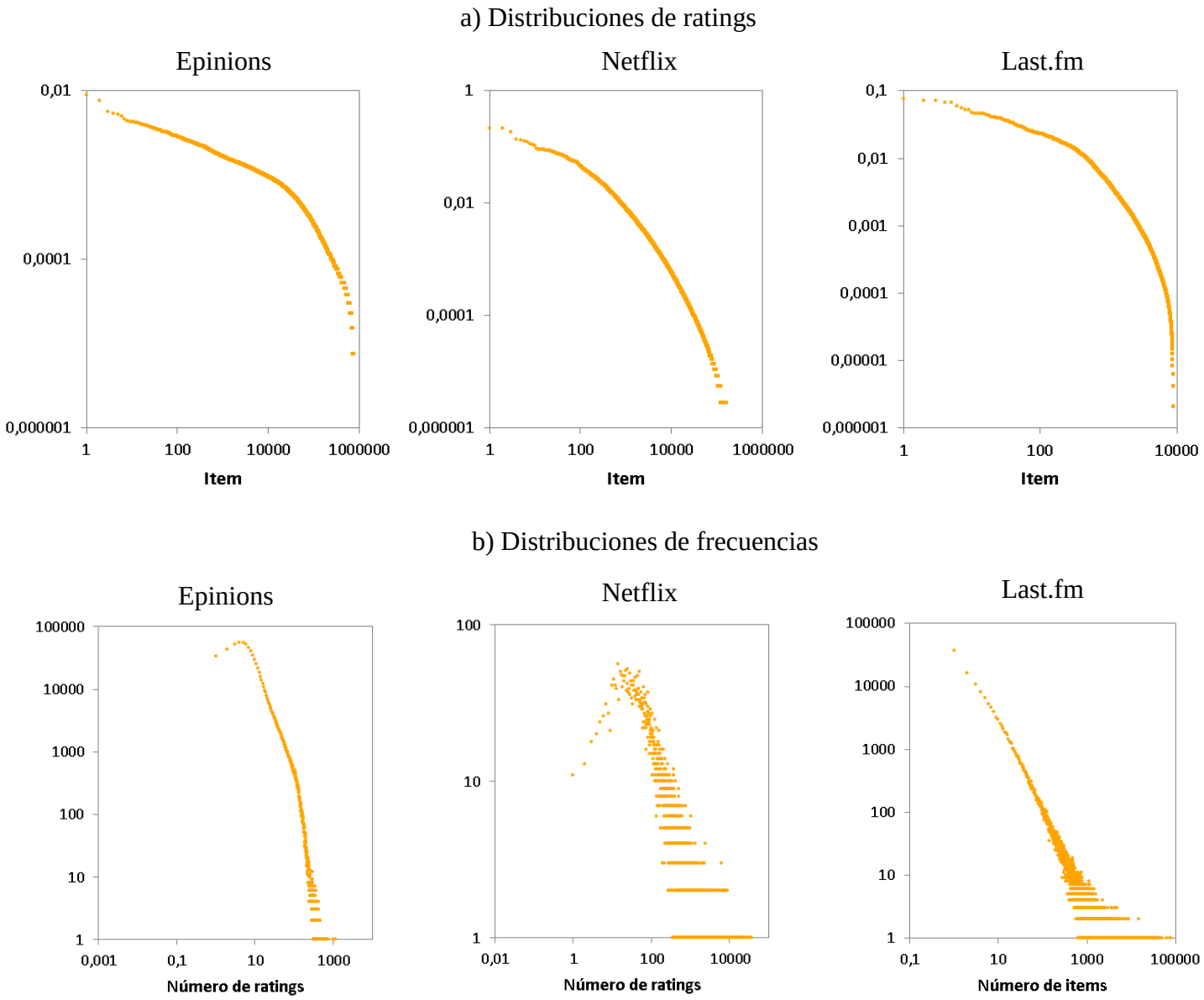


Figura 6. Distribución de ratings y de frecuencias de Epinions, Netflix y Last.fm.

- Manifestación: Una vez que han formado su opinión acerca de un ítem, los usuarios pueden manifestarla o no y, en caso de hacerlo, lo hacen de muy diversas formas.

  Los datasets públicos de recomendación muestran que los hábitos de los usuarios a la hora de emitir votos u opiniones están sesgados hacia, generalmente, los casos positivos. En (Steck 2011) muestran la importancia que ello tiene en distorsionar el conocimiento que toman como entrada los recomendadores y los métodos que los evalúan.

En este trabajo, se asume que la generación de gustos ya se ha producido y éstos están fijos. Nos centramos, por tanto, en estudiar cómo se generan y distribuyen las manifestaciones de estos gustos en forma de ratings. Esto es, cómo los usuarios valoran los ítems mediante un dato numérico que representa sus gustos.

2.4 Modelo de formación de opinión

El precedente más cercano del estudio que abordamos aquí es un trabajo preliminar presentado en un taller del congreso RecSys 2012 (Blattner 2012). En este trabajo se presenta un modelo de formación de opiniones cuyos objetivos son similares a los nuestros, esto es, imitar la distribución que siguen los ratings reales de forma que el modelo pueda ser utilizado en un futuro para generar los datos de entrada de los recomendadores y poder así evaluarlos.

Basándose en estudios anteriores que afirman que la opinión de nuestros amigos influye muy notablemente en la nuestra, los autores del artículo proponen un modelo que tiene esta influencia en cuenta y que se compone principalmente de los siguientes tres ingredientes:

- Red de influencias: grafo en el cual los nodos son los individuos y las conexiones entre ellos indican que uno influye en el otro.
- Opinión inicial de cada usuario sobre cada ítem: indica, mediante un valor numérico, lo que cada usuario piensa de cada ítem cuando todavía no se ha intercambiado información mediante la red de influencia. Representa el propio criterio personal del usuario junto con la influencia de otras fuentes externas a la red social: publicidad, críticas, etc.
- Dinámicas de influencia: marcan la interacción entre los usuarios y cómo unos influyen en la opinión de otros. Para cada ítem, se genera la opinión de todos los usuarios acerca de él de la siguiente forma:
  - Cuando le toca el turno a un usuario, mira cuantos de sus vecinos tienen una opinión positiva acerca del ítem y altera su opinión de acuerdo a ese número. Si dicha opinión supera un cierto límite, se considera que el ítem le gusta lo suficiente y se marca como voto positivo.
  - Los usuarios se recorren de forma aleatoria y, una vez que han formado su opinión (sea positiva o no) ya no pueden modificarla, es decir, no vuelven a intervenir.

Aunque los autores del artículo comparten el mismo objetivo que nosotros, el modelo que plantean difiere en varios aspectos:

- Blattner et al se centran en cómo se genera la opinión, identificando opinión con voto, mientras que nosotros consideramos opiniones ya formadas, y pondremos el foco en cómo se genera la oportunidad de opinar. Incorporamos otras variables como el descubrimiento, sin el cual no se puede llegar a realizar un voto, pues no se conoce el ítem.
- Blattner et al consideran una comunicación instantánea ya que una vez que le toca el turno al usuario, éste se ve influenciado por todos los vecinos que tienen una opinión positiva del asunto, lo cual implica que se comunica con todos ellos. Es decir, asumen que la opinión de los vecinos influye, pero no estudian las distintas dinámicas de comunicación.

  En nuestro caso, modelizamos el proceso de comunicación introduciendo probabilidades que determinan si dicha comunicación se produce o no y considerando distintas variantes en la forma de comunicar: hablar con varios vecinos o sólo con uno, informar de varios ítems o sólo de uno, etc.

3. Diseño del modelo

Como ya se ha explicado anteriormente en la sección 1.2 de la introducción, el objetivo del proyecto es diseñar un escenario que permita estudiar tanto la generación y distribución de ratings como la propagación de información en redes sociales. En este capítulo se exponen, en base a estos dos objetivos, qué elementos y dinámicas forman parte del modelo.

Antes de entrar en detalles, cabe destacar que el modelo diseñado pretende alcanzar un equilibrio entre sencillez, generalidad y realismo, para lo cual se ha realizado una selección intencionada de las variables cuya relación queremos estudiar (relevancia, descubrimiento, rating…) y de las dinámicas específicas que pretendemos analizar.

3.1 Planteamiento general

En primer lugar, cabe destacar que se ha elegido el dominio de las películas para contextualizar el escenario diseñado. Sin embargo, el modelo permite abstraerse del tipo de ítem que se está tratando. De hecho, en la sección 6.3 de los experimentos se explican ajustes de datos de ratings reales realizados sobre objetos de muy diverso tipo.

Veamos, a continuación, los elementos que se han incorporado al modelo para modelizar el proceso de generación de ratings. Evidentemente, la red social y la lista de ítems, en nuestro caso, películas, son indispensables. El resto de elementos son una serie de interacciones usuario-ítem y usuario-usuario que observamos al analizar el proceso mediante el cual un usuario consigue generar un rating.

Descubrimiento

En primer lugar, para que un usuario pueda generar un rating sobre un ítem es necesario que lo haya descubierto previamente. Este descubrimiento puede llevarse a cabo a través de diversos medios:

- Red social: Una de las formas más comunes mediante la cual obtenemos información acerca de un cierto ítem es cuando un amigo nos informa acerca de él. Por tanto, el descubrimiento a través de la red social se produce cuando los usuarios intercambian información acerca de los ítems de forma que algunos de ellos descubren ítems que no conocían anteriormente.
- Fuentes de descubrimiento exógenas: Cuando estamos interesados en descubrir información acerca de nuevas películas, libros, música… no sólo acudimos a nuestros amigos, sino que también buscamos información en otras fuentes muy diversas (televisión, internet, periódicos…) que tienen como finalidad generar dicha información e introducirla en el sistema.

  En escenarios tradicionales es común que las fuentes de descubrimiento sean externas a la red social (prensa, guías especializadas, páginas web, publicidad, recomendadores...) aunque en los nuevos escenarios de medios sociales es cada vez más habitual que sean los propios usuarios quienes crean la información –por ejemplo, en Twitter los tweets son generados por la propia red social.

En general, podemos considerar cuatro grupos en los que clasificar este tipo de fuentes:

- Buscadores: se caracterizan por el hecho de que es el usuario el que acude a ellos en busca de información. Esto hace que sus resultados muestren una tendencia o sesgo hacía los gustos del usuario.
- Publicidad: a diferencia de los buscadores, la publicidad ofrece información al usuario sin que éste así lo requiera y tiene por objetivo intentar incrementar el consumo de aquellos ítems que publicita.
- Recomendadores: al igual que la publicidad, ofrecen información por iniciativa propia. Sin embargo, lo que los caracteriza es intentar informar al usuario de aquello que más le puede interesar, basándose en su actividad previa.
- Encuentro fortuito: se produce, por ejemplo, al cambiar de canal en la televisión o al consultar la cartelera. En este tipo de descubrimiento, al contrario que lo que ocurre con los anteriores, podemos considerar que todos los ítems tienen la misma probabilidad de ser descubiertos, si ignoramos el aspecto temporal.

El descubrimiento, tanto por red social como por fuentes de descubrimiento externas, es esencialmente fortuito, pues aunque el usuario puede elegir el medio por el que informarse, no puede determinar de antemano los ítems acerca de los cuales va a obtener información. Pese a ello, se puede considerar una cierta tendencia a encontrar ítems que nos interesan, pues tendemos a relacionarnos con gente que tiene gustos similares a los nuestros o buscamos información en lugares que sabemos suelen mostrar cosas que nos gustan, y nos servimos de la capacidad de los buscadores para encontrar información que satisface el interés que expresamos mediante una consulta.

Consumición

Una vez descubierto, el usuario puede decidir consumir el ítem. Por ejemplo, si es una película puede verla, si es un libro, leerlo, si es una canción, escucharla, etc. Mientras no consuma el ítem, el usuario no podrá votarlo, pues desconoce si le ha gustado o no.

A la hora de decidir acerca de ver o no una película intervienen varios y muy diversos factores, tales como la opinión de los amigos, las críticas, la información que poseamos acerca de dicha película, el lugar donde la hayamos descubierto, etc. La influencia de cada uno de estos factores varía, además, en función del canal por el que se considere hacer la consumición:

- Cine: si pensamos en ir al cine a ver la película, solemos estar más influenciados por las críticas, nuestros amigos y la publicidad. Además, en este canal, el tiempo entre descubrimiento y consumición es menor, pues está limitado por el tiempo que la película está en cartelera.
- Alquiler/compra: si estamos considerando alquilar/comprar la película, nos suelen importar más los datos de dicha película (género, director, actores…) y tenemos más tiempo entre descubrimiento y consumición para considerar la decisión.
- Televisión: por último, las películas que consideremos ver en la televisión dependen fundamentalmente de la frecuencia de iteración del usuario con este medio y de la distribución de películas en él.

Sin embargo, en el modelo vamos a ignorar estos factores y a asumir un único canal de consumición. También se considerará que no poseemos ninguna información de antemano sobre la película que condicione su elección sobre el resto.

Votación

Tras consumir el ítem, el usuario ya está en condiciones de decidir si quiere o no votarlo. Esta decisión, al igual que el resultado de la votación, suele depender de si le ha gustado o no. De hecho, los datasets públicos muestran que los hábitos de los usuarios a la hora de emitir votos suelen estar sesgados hacia los casos positivos (Steck 2010, 2011).

Además de los gustos de los usuarios se podrían considerar otras influencias (nuevamente amigos, críticas…) pero no las incluimos como parte del modelo por simplicidad.

Comunicación

Una última decisión, necesaria para la propagación de la información en la red social, que debe tomar el usuario después de consumir el ítem y con independencia de si lo ha votado o no, es si quiere comunicar a sus conocidos la información de dicho ítem.

Al igual que ocurría con la votación, la comunicación está fuertemente ligada a la opinión personal del usuario acerca de dicha película. Cabe pensar que tenemos la tendencia a comunicar a nuestro entorno con más probabilidad aquello que más nos gusta, si bien puede resultar también de interés en ocasiones comunicar experiencias negativas.

Relevancia

Tal y como se ha explicado anteriormente, las decisiones sobre votar y comunicar están comúnmente condicionadas por los gustos e intereses del usuario, pues dependen de su opinión acerca del ítem consumido. Incorporamos por ello al modelo estos gustos de los usuarios sobre los ítems, a los que denominaremos relevancia, tomando esta denominación del campo de la recuperación de información.

Aunque se podrían considerar modelos en los que los gustos cambiaran con el tiempo en función de factores como la opinión de los amigos, las críticas, la experiencia… este modelo se va a caracterizar porque las opiniones de los usuarios permanecen constantes a lo largo de toda la simulación.

Establecer una relevancia variable iría más acorde con un proyecto en el que observar la formación, fluencias y propagación de estados de opinión, gustos y relevancias fuera uno de los principales objetivos. En nuestro caso estamos más interesados en estudiar su efecto en el resto de variables aleatorias, y por tanto introducir una relevancia cambiante sólo complicaría dicho estudio.

En resumen, para generar ratings sobre los ítems debemos incluir en el escenario actividades de descubrimiento y decisiones acerca de consumir, votar y comunicar un ítem así como reproducir los gustos de los usuarios respecto a los distintos ítems.

Las actividades de descubrimiento y comunicación de la información enlazan directamente con el segundo objetivo del proyecto, esto es, el estudio de la propagación y el descubrimiento de información en redes sociales. De esta forma tomamos un escenario de recomendación y generación de ratings como campo particular de estudio de los fenómenos de propagación.

Veamos a continuación una definición más formal del modelo, de sus parámetros y variables y de las simplificaciones y las decisiones que se han tomado para elaborarlo.

3.2 Definición formal

En este apartado se describe un modelo cuyo objetivo es capturar un escenario del tipo descrito anteriormente con simplificaciones necesarias para hacerlo tratable (teórica y empíricamente), pero buscando una generalidad que lo haga interesante y permita extraer un análisis cercano a la realidad.

La interacción entre usuarios e ítems la modelamos mediante cinco variables aleatorias: *relevance* (relevancia), *find* (descubrimiento), *watch* (consumición), *rate* (votación) y *tell* (comunicación), que se corresponden con cada una de las interacciones usuario-ítem y usuario-usuario que hemos descrito al inicio de este capítulo. Dado que el modelo se ha implementado en inglés, en ocasiones a lo largo del documento escribiremos ciertos términos en este idioma, en cursiva.

Definimos todas las variables aleatorias como booleanas, es decir, toman un valor de 0 o 1 en función de si el ítem es relevante para el usuario o ha sido descubierto, consumido, votado o comunicado por el mismo.

En la Tabla 1 se resumen estas variables aleatorias que modelizan el escenario que queremos representar.

Variable	Descripción
$relevance\ (u,i)$	Indica si el ítem i es relevante para el usuario u.
$find(u,i)$	Indica si el ítem i ha sido descubierto por el usuario u.
$watch(u,i)$	Indica si el usuario u ha visto el ítem i.
$rate(u,i)$	Indica si el usuario u ha votado el ítem i.
$tell(u,v,i)$	Indica si el usuario u le ha hablado al usuario v acerca del ítem i.

Tabla 2. Lista de las variables aleatorias del modelo con sus correspondientes descripciones.

A continuación analizamos en detalle cada una de estas variables.

3.2.1 Relevancia

La relevancia representa los gustos de los usuarios sobre los distintos ítems. Así, dado un usuario y un ítem, *relevance* tomará un valor de 1 si el ítem es relevante para el usuario y 0 en caso contrario.

Como se ha explicado anteriormente, esta variable no depende del tiempo y permanece constante a lo largo de toda la simulación.

3.2.2 Descubrimiento

Tal y como se explicó al inicio de esta sección, el descubrimiento puede realizarse por dos vías, la red social y las fuentes externas de descubrimiento. Estas últimas engloban, a su vez, fuentes tales como los buscadores, la publicidad, el encuentro aleatorio o los recomendadores.

En realidad, podríamos generalizar y abstraer un poco más el concepto de recomendador, por cuanto a su papel de fuente de información respecta, englobando por ejemplo la publicidad como un recomendador no personalizado que sigue una distribución no uniforme, esto es, no todos los ítems tienen la misma probabilidad de ser publicitados. El encuentro fortuito se modelizaria de la misma forma que la publicidad, pero considerando una distribución uniforme. Lo mismo podemos hacer, a un nivel de efecto estadístico global, con los buscadores, abstrayéndonos de las consultas individuales, y considerando que podrían ser representados por un recomendador sesgado a la relevancia de los ítems para el usuario.

Podemos considerar, por tanto, que las fuentes de descubrimiento son un conjunto $D = \{d_i\}_{i=1}^{n}$ de recomendadores. Cada d_i tiene adjudicada una probabilidad $p(d_i)$ que determina la probabilidad con la que dicho recomendador es elegido como fuente de descubrimiento. Esta probabilidad es un parámetro del modelo y se debe cumplir que:

$$\sum_{i=1}^{n} p(d_i) = 1$$

El descubrimiento producido por la red social también podría modelizarse como un recomendador, pues se puede interpretar que un amigo que nos habla de películas actúa como un recomendador humano. Sin embargo, los parámetros de este recomendador (probabilidad de ser elegido, distribución con la que se elige la película a recomendar…), se establecen de forma diferencial en nuestro modelo respecto de los demás recomendadores, pues nos interesa estudiar con más detalle el efecto de las dinámicas de red que determinan qué usuario actúa sobre quién en cada momento. Por este motivo no modelizaremos a los usuarios como recomendadores sino como otro tipo de agente.

El descubrimiento por vía de la red social depende de la forma de comunicación y por ello se verá con más detalle cuando analicemos esta variable.

Un último aspecto relevante de la modelización del proceso de descubrimiento es que si el usuario descubre una película que ya ha sido descubierta, y con independencia de si se ha consumido o no, se ignora dicho descubrimiento. Es decir, por simplificación se considera que las películas sólo se descubren una vez.

3.2.2.1 Velocidad de descubrimiento por fuentes externas

Para controlar la velocidad de descubrimiento mediante fuentes externas se utiliza un ratio entre películas descubiertas y vistas, un parámetro del modelo que relaciona descubrimiento y consumición.

Veremos a continuación, al explicar la consumición, que estableceremos el número de películas que se consumen por unidad de tiempo como un número fijo. Así, lo que conseguimos al introducir un ratio entre películas encontradas y vistas es controlar las encontradas pues las vistas son fijas.

Si el ratio es muy bajo apenas se descubrirá nada externamente y la propagación de información vendrá dada casi exclusivamente por la red social. Si por el contrario aumentamos el ratio, los usuarios empezarán a descubrir independientemente de su interacción con otros usuarios y el descubrimiento exógeno eclipsará la propagación en la red social.

3.2.3 Consumición

Tras descubrir los ítems, el usuario debe decidir si quiere consumirlos o no. La forma de modelizar este proceso es considerar el conjunto de películas descubiertas (y no consumidas) por el usuario y elegir una cierta cantidad de ellas para ser consumidas. Una vez elegidas, éstas se eliminan del conjunto de películas descubiertas y no consumidas.

En la elección se realizan dos asunciones:

- Probabilidad uniforme: todas las películas tienen la misma probabilidad de ser elegidas para consumir. Esto implica que las posibles influencias de la relevancia a la hora de consumir (opinión de los amigos, críticas, información acerca de la película, lugar donde se haya descubierto…) no se consideran en el modelo.
- Se elige un número fijo de películas para ser consumidas. Este valor es un parámetro del modelo y es importante tener en cuenta que sólo es posible consumir dicho número si se ha descubierto una cantidad igual o superior. En caso contrario, se consumirán todas las que se hayan descubierto.

  Se podría haber modelizado al contrario, considerando las películas descubiertas como un número fijo y las vistas determinarlas en función de un cierto ratio, o incluso una función no lineal, del número de películas descubiertas. Sin embargo, parece más realista asumir que vemos un número fijo de películas por unidad de tiempo, por ejemplo a la semana, mientras que las que descubrimos presentan más variación. Así, si estamos buscando una película para ver, tenemos claro que sólo queremos ver una, pero no sabemos cuántas descubriremos hasta que demos con una que nos guste.

El hecho de considerar únicamente aquellas películas que no han sido consumidas con anterioridad implica que, para simplificar, consideramos que una película sólo puede ser consumida una única vez.

3.2.4 Votación

La votación es la forma en la que el usuario manifiesta su opinión acerca de un ítem. Es necesario, por tanto, que lo haya descubierto y consumido previamente. Además, para simplificar, circunscribimos la decisión de votar al instante inmediatamente después de consumir la película, en el mismo proceso. Esto implica que si se decide no votar, ya no se realizará la votación en ningún otro momento, pues no se puede volver a consumir dicha película.

Respecto a la representación de los votos, es común en el campo de los sistemas de recomendación, así como en los entornos donde los usuarios valoran productos (películas, hoteles, etc.) utilizar escalas numéricas p.e. de 0 a 5, o valores binarios (me gusta / no me gusta). En nuestro modelo consideramos esta última opción, pues el grado de relevancia, más allá de la consideración binaria, no aporta un matiz particularmente importante para nuestro análisis.

Es importante, por otra parte, no confundir el valor de rating con el valor de la variable aleatoria *rate*. Esta variable indica si un usuario ha votado o no un ítem, pero no dice nada del valor de dicho voto.

3.2.4.1 Influencia de la relevancia

Los conjuntos de datos públicos sobre recomendación muestran que la decisión sobre si votar o no un ítem está condicionada por la relevancia de dicho ítem para el usuario

(Steck 2010, 2011), de hecho, en estos conjuntos se observa que votamos más los ítems que nos gustan que los que no nos gustan. Por ello, para modelizar este proceso de decisión consideramos dos parámetros, $p(rate|R)$ y $p(rate|\neg R)$ que representan, respectivamente, la probabilidad de votar dado que el ítem es relevante y dado que no lo es.

Una vez que hemos decidido votar, nuestro voto está unívocamente determinado por la relevancia y no hay ninguna probabilidad que medie ningún otro factor en el valor de voto: si el ítem es relevante para el usuario, el voto es 1 y si no lo es, el voto es 0.

3.2.5 Comunicación

La comunicación es el proceso mediante el cual dos usuarios intercambian información acerca de los ítems que han consumido. Este proceso es modelizado por la variable *tell* que indica si un usuario está transmitiendo información a otro/s o no.

Al igual que ocurre con la votación, la comunicación está fuertemente ligada a la relevancia por lo que volvemos a considerar dos parámetros, $p(tell|R)$ y $p(tell|\neg R)$, que determinan la probabilidad de que un usuario hable acerca de una película sabiendo que es (o no) relevante para él.

En la Figura 7 podemos observar un resumen del modelo descrito en este capítulo.

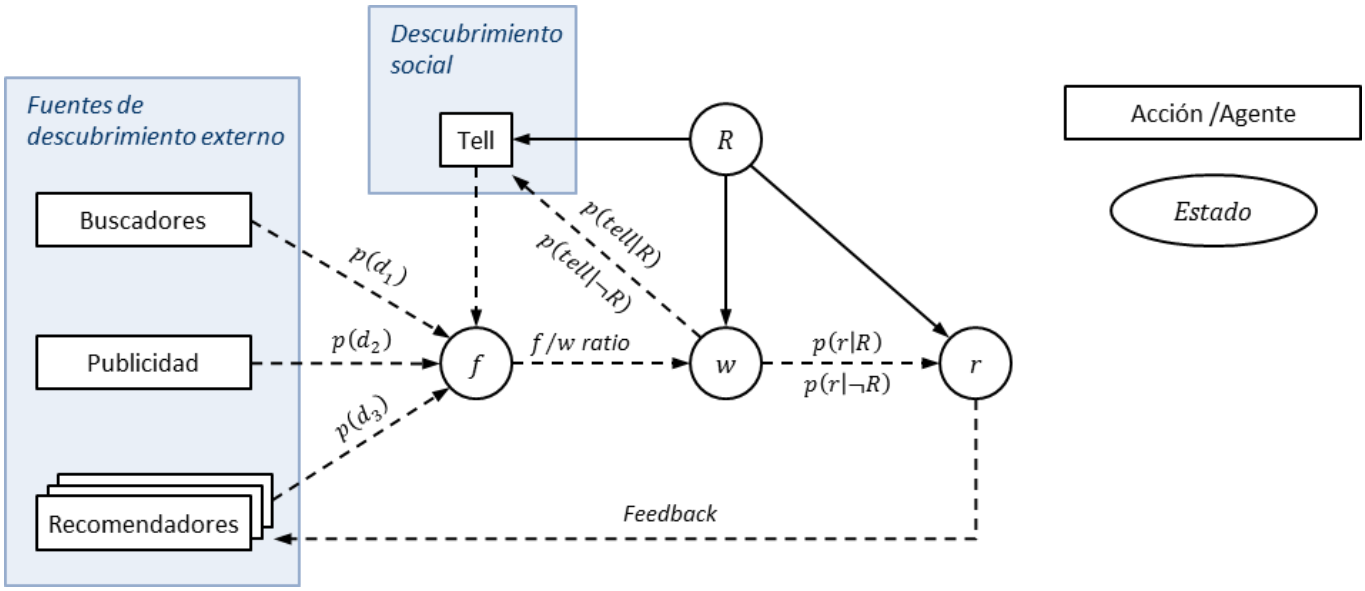


Figura 7. Esquema del modelo propuesto.

3.2.6 Dependencias entre las variables

Enumeramos y resumimos a continuación las variables aleatorias que definimos en nuestro modelo y las hipótesis y simplificaciones consideradas, expresadas en términos de las dependencias entre variables.

find

Una película no puede ser descubierta dos veces, es decir, dado un usuario u, un ítem i y dado que dicho ítem ha sido encontrado por el usuario, la probabilidad de volver a hacerlo es nula: $p(f|f,i,u) = 0$

Si el usuario ha consumido, votado o transmitido la película, entonces también ha tenido que descubrirla $p(f|w,i,u)=1$, $p(f|r,i,u)=1$ y $p(f|t,i,u)=1$.

watch

Si una película ya ha sido vista, no se puede volver a ver: $p(w|w,i,u)=0$

Para ver una película es necesario haberla descubierto previamente: $p(w|\neg f,i,u)=0$

La probabilidad de ver una película que ya ha sido descubierta pero no consumida es uniforme:

$$p(w|\neg w,f,i,u)=\frac{1}{|F_u|}$$

donde F_u es el conjunto de películas descubiertas y no consumidas por el usuario u.

Si el usuario ha votado o transmitido la película, es necesario que la haya consumido previamente: $p(w|r,i,u)=1$ y $p(w|t,i,u)=1$.

rate

Las películas sólo se pueden votar una vez: $p(r|r,i,u)=0$

Para votar una película es necesario haberla descubierto y consumido anteriormente: $p(r|\neg f,i,u)=0$ y $p(r|\neg w,f,i,u)=0$

La decisión sobre si votar o no está condicionada por la relevancia del ítem para el usuario, se consideran los parámetros $p(rate|R)$ y $p(rate|\neg R)$.

tell

Para transmitir una película es necesario haberla descubierto y consumido con anterioridad: $p(t|\neg f,i,u)=0$ y $p(t|\neg w,f,i,u)=0$

La decisión sobre si comunicar o no una película depende de la relevancia de dicha película para el usuario, se consideran los parámetros $p(tell|R)$ y $p(tell|\neg R)$.

relevance

La relevancia es constante a lo largo del tiempo y no depende de ninguna otra variable.

En la Tabla 3 se resumen estas dependencias entre las distintas variables:

	find	***watch***	***rate***	***tell***	***relevance***
find	$p(f\|f,i,u)=0$	$p(f\|w,i,u)=1$	$p(f\|r,i,u)=1$	$p(f\|t,i,u)=1$	
watch	$p(w\|\neg f,i,u)=0$	$p(w\|w,i,u)=0$	$p(w\|r,i,u)=1$	$p(w\|t,i,u)=1$	
rate	$p(r\|\neg f,i,u)=0$	$p(r\|\neg w,f,i,u)=0$	$p(r\|r,i,u)=0$		$p(rate\|R)$ y $p(rate\|\neg R)$
tell	$p(t\|\neg f,i,u)=0$	$p(t\|\neg w,f,i,u)=0$			$p(tell\|R)$ y $p(tell\|\neg R)$

Tabla 3. Dependencias entre las distintas variables.

4. Marco de simulación

El modelo descrito en el apartado anterior permite definir patrones de comportamiento, a partir de los cuales es posible definir simulaciones de la evolución del sistema formado por los usuarios, el flujo de comunicación entre ellos, sus acciones y los datos (ratings) que se generan en el proceso. En esta sección se describe la estructura, configuración y puesta a punto de un sistema de simulación basado en este modelo, así como los parámetros que recibe. Dado que la aplicación se ha escrito en inglés, los nombres de los parámetros también se presentan en este idioma, en cursiva y entre paréntesis.

El sistema de simulación consta de una serie de usuarios y de ítems entre los que se producen las interacciones descritas en el modelo: descubrimiento, consumición, comunicación y votación. Para llevar a cabo estas interacciones es necesario definir en el proceso de simulación una serie de elementos que no vienen definidos por el modelo, porque se refieren a aspectos técnicos e implementativos. Así, hay que especificar las entradas y salidas, las inicializaciones necesarias, el orden en el que se suceden los eventos (intervenciones de los usuarios, descubrimientos, consumiciones, votaciones), los momentos en los que se samplea de las distribuciones definidas en el modelo, las distintas formas o modos de comunicación, los parámetros y distribuciones propias que se derivan de la modelización de dichos modos de comunicación ($p(start)$, ($p(go)$…)

En particular, es necesario representar la relevancia, esto es, lo gustos de los usuarios, pues son parte fundamental del modelo y median en la toma de decisiones. Sin embargo, la relevancia es un dato prácticamente imposible de obtener de la realidad cuando hay miles o millones de ítems pues, aunque los usuarios se prestasen a manifestar su opinión – que no es el caso – tendrían que probar todos los ítems. Por ello, la relevancia de los usuarios se simulará al inicio de la simulación.

4.1 Inicialización de la simulación

La simulación toma como uno de sus parámetros una red social arbitraria. Ésta puede tomarse de cualquier conjunto de datos disponible, o bien generarse desde el propio programa utilizando modelos de grafos aleatorios, dados los parámetros de éstos (típicamente número de usuarios y número total o promedio de conexiones). A partir de la red, la primera acción de la simulación es inicializar la distribución de relevancia, que indica qué ítems son relevantes para qué usuarios.

4.1.1 Tipos y construcción de la red social

Como ya se explicó anteriormente en la sección 2.1.1 del estado del arte, una red social se representa comúnmente mediante un grafo no ponderado que puede ser dirigido o no. Este grafo es un parámetro del sistema (*Graph*) que puede tomarse como dato externo o ser generado por la propia simulación.

El soporte y recorrido de la mayoría de grafos, incluido los externos, se lleva a cabo mediante la librería *Jung*[5] de Java que ofrece una gran variedad de métodos para

[5] http://jung.sourceforge.net/

generar y modelizar grafos. También la generación, en caso de que se opte por esta opción, se lleva a cabo mediante dicha librería.

Se tienen, por tanto, tantos modelos de formación de grafos como ofrece Jung – Barabási, Erdös, Kleinberg y Eppstein – los cuales vienen determinados por el número de usuarios (*Nr. of Users*) y el grado promedio (*Average degree*), ambos parámetros del sistema. Alternativamente, también se ofrece la opción de trabajar con un grafo completo, pero la implementación de este tipo de grafo se ha realizado de forma independiente a Jung para aumentar la eficiencia.

Por último, el hecho de que la red social se pueda tomar de un fichero externo permite ejecutar la simulación sobre redes sociales reales (Facebook, Twitter…), pero también sobre grafos creados por cualquier algoritmo de generación automática de cualquier librería, como por ejemplo *SNAP*[6] – librería de C++ utilizada durante los experimentos para la generación de grafos de forma externa a la simulación.

4.1.2 Distribución de relevancia

La relevancia es una relación binaria entre ítems y usuarios que se observa únicamente cuando se produce un rating. Si no se ha realizado ningún voto, puede ser que el mismo usuario no conozca su opinión porque no ha descubierto el ítem o que sí la conozca pero que haya decidido no expresar su voto. Por tanto, no es en general posible conocer los gustos de los usuarios de forma exhaustiva y por ello nos vemos en la necesidad de simularlos.

En escenarios reales se ha observado que la relevancia no se distribuye de manera uniforme: hay ítems que gustan a muchos usuarios, otros que no gustan prácticamente a ninguno, y la mayoría de ellos gustan a unos pocos. Esto nos lleva a aventurar que la relevancia sigue una distribución desigual de tipo Pareto. Validar la hipótesis, o incluso tratar de derivar una distribución real, se podría hacer mediante algún sondeo crowsourced como trabajo futuro.

Uno de los modelos más sencillos y clásicos para este tipo de distribuciones sesgadas de tipo Pareto es la función power law definida por la siguiente fórmula:

$$frec_{\alpha}(k) = \beta k^{-\alpha}$$

donde $frec_{\alpha}(k)$ indica el número de usuarios para los que el ítem k es relevante, siendo $k = 1,2,\ldots$ el orden de los ítems por $frec_{\alpha}(k)$. En nuestro caso α es un parámetro de la simulación (*Relevance alpha*) y β una constante cuyo valor se deriva del resto de parámetros como mostraremos a continuación.

Teniendo en cuenta que el priori de relevancia $p(rel)$ indica la probabilidad de que, al escoger un par usuario-ítem al azar, el ítem sea relevante para el usuario, se debe cumplir:

$$\sum_{k} frec_{\alpha}(k) \sim p(rel)\mathcal{U} \cdot \mathcal{I}$$

donde $\mathcal{U}$ y $\mathcal{I}$ son, respectivamente, el número de usuarios (*Nr. of users*) y el número de ítems (*Nr. of items*). Ambos valores son parámetros del sistema.

Así, podemos despejar β como:

[6] http://snap.stanford.edu/snap/

$$\beta = \frac{p(rel) \cdot \mathcal{U} \cdot \mathcal{I}}{\sum_k k^{-\alpha}}$$

En la Figura 8.a podemos ver la forma de esta función para distintos valores del parámetro α cuando se tienen 4000 usuarios, 4000 ítems y un priori $p(rel) = 0.2$.

a) power law clásica b) power law modificada

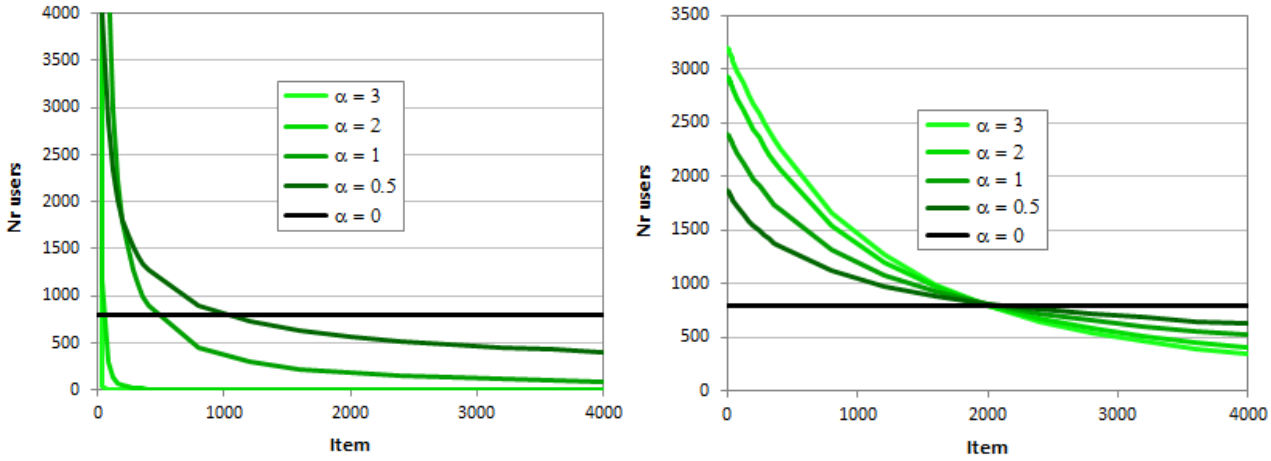


Figura 8. Forma de la función power law (clásica y modificada) para distintos valores del parámetro α.

Observamos que α determina lo pronunciada que es la curva. Cuando es 0, obtenemos una recta constante, mientras que en el caso de $\alpha = 3$, la curva está prácticamente pegada a los ejes. Sin embargo, este cambio es demasiado brusco. Para $\alpha \geq 0.5$, los ítems más relevantes sobrepasan los 4000 usuarios y para $\alpha \geq 2$, la mayor parte de los ítems no son relevantes para ningún usuario. Por ello utilizamos una versión alternativa de la función power law que introduce un desplazamiento en ambos ejes:

$$frec_\alpha(k) = c_1 + \beta(k + c_2)^{-\alpha}$$

Además, imponemos como restricción que el mínimo y el máximo se alcancen en los extremos y que estén acotados por 0 y $\mathcal{U}$ respectivamente:

$$0 \leq frec_\alpha(\mathcal{I}) \leq frec_\alpha(k) \leq frec_\alpha(0) \leq \mathcal{U}$$

Este máximo y mínimo son funciones de α cuyo comportamiento podemos establecer para valores extremos de α.

$$\lim_{\alpha \to 0} \min(\alpha) = p(rel)\mathcal{U} \qquad \lim_{\alpha \to \infty} \min(\alpha) = 0$$

$$\lim_{\alpha \to 0} \max(\alpha) = p(rel)\mathcal{U} \qquad \lim_{\alpha \to \infty} \max(\alpha) = \mathcal{U}$$

Existen numerosas funciones que cumplen estos límites, pero vamos a utilizar las siguientes por su simplicidad:

$$\min(\alpha) = \mathcal{U}\frac{p(rel)}{1+\alpha} \qquad \max(\alpha) = \mathcal{U}\frac{p(rel)+\alpha}{1+\alpha}$$

De esta forma, se obtiene el siguiente sistema de ecuaciones:

$$\left.\begin{aligned} \sum_k c_1 + \beta(k + c_2)^{-\alpha} &= p(rel)\mathcal{U} \cdot \mathcal{I} \\ c_1 + \beta c_2^{-\alpha} &= \max(\alpha) \\ c_1 + \beta(\mathcal{I} + c_2)^{-\alpha} &= \min(\alpha) \end{aligned}\right\}$$

donde α, $p(rel)$, $\mathcal{U}$ y $\mathcal{I}$ son parámetros conocidos por el sistema y c_1, c_2 y β son las incógnitas que se quieren calcular.

Podemos despejar c_1 y β en función de c_2 como:

$$\beta = \frac{\max(\alpha) - \min(\alpha)}{c_2^{-\alpha} - (\mathcal{I} + c_2)^{-\alpha}}$$

$$c_1 = \max(\alpha) - \beta c_2^{-\alpha}$$

y c_2 se calcula inicializando su valor como $c_2 = 0$ e incrementándolo en 1 hasta que:

$$\sum_k c_1 + \beta(k + c_2)^{-\alpha} > p(rel)\mathcal{U} \cdot \mathcal{I}$$

Con estos ajustes, en la Figura 8.b podemos observar cómo las curvas se encuadran dentro de los límites $1 \leq frec_\alpha(k) \leq \mathcal{U}$ y cómo se ha conseguido suavizar la pendiente.

Una vez establecido el número de usuarios que consideran relevante cada ítem, se decide a qué usuario concreto le gusta cada ítem muestreando uniformemente entre todos los usuarios, cumpliendo el número necesario para cada ítem.

4.2 Bucle central de la simulación

El bucle central de la simulación recorre el conjunto de usuarios de forma aleatoria o secuencial. Se considera que ha transcurrido una unidad de tiempo cuando se han recorrido $q \cdot \mathcal{U}$ usuarios, donde q es un parámetro de la simulación (*Fraction of users per time step*). Si se realiza el recorrido aleatorio, no todos los usuarios intervienen generalmente en una unidad de tiempo (no es imposible pero sí muy improbable), porque al ser recorridos aleatoriamente puede haber usuarios que intervengan más de una vez y otros que no intervengan ninguna. Si se recorren secuencialmente y $q = 1$, entonces una unidad de tiempo coincide precisamente con la intervención de todos los usuarios una única vez.

Un usuario, en su turno, realiza las siguientes acciones, en el orden en que se enumeran:

- Descubrimiento: el usuario descubre un cierto número de películas mediante las distintas fuentes externas de descubrimiento, modelizadas todas ellas como recomendadores.
- Consumición: el usuario elige entre las películas descubiertas las que va a ver.
- Votación: para cada película consumida, y únicamente tras consumirla, el usuario decide si introduce o no un voto para dicha película. En caso de realizar la votación, el usuario asignará un voto relevante (positivo) si la película es relevante para dicho usuario o un voto no relevante (negativo) si no lo es.

- Comunicación: después de consumir, decide si contar o no a sus amigos las películas vistas.

A continuación se analizan cada una de estas acciones, explicando cómo se simulan y los parámetros que las modelizan.

4.2.1 Descubrimiento

En cada turno, un usuario descubre un cierto número de películas por vía exógena (es decir, fuera de su red social). Este número es variable pero gira en torno a una cierta media. El valor se modeliza mediante una distribución de Poisson cuya media λ viene determinada por la siguiente relación:

$$r = \frac{\lambda}{w}$$

donde w es el número de películas consumidas por turno (*Nr. watched movies per step*) y r es el ratio entre películas descubiertas y vistas (*Found/watched ratio*). Ambos valores r *y* w son parámetros de la simulación y su valor está fijo, por lo que para calcular λ únicamente multiplicamos r *y* w.

El motivo por el que se ha elegido la distribución de Poisson para modelizar el descubrimiento es que esta distribución expresa, a partir de una frecuencia de ocurrencia media, la probabilidad de que ocurra un determinado número de eventos durante cierto periodo de tiempo. En nuestro caso, los eventos son descubrimientos y el período de tiempo, un turno del usuario.

Es posible asignar al parámetro r el valor 0. En este caso, se elimina el descubrimiento exógeno ($\lambda = 0$) y toda la propagación recae sobre la red social. Para permitir este tipo de situaciones y que la simulación pueda comenzar, se ha añadido un nuevo parámetro (*Initial nr. of users that know each item*) que establece el número de usuarios conocedores de alguna película desde el principio. Estos usuarios se eligen al inicializar la simulación, y su número debe ser mayor que cero cuando no se contempla el descubrimiento exógeno para que la simulación se inicie y progrese.

Una vez calculado el número de películas que se van a descubrir, se determina el recomendador por el que se encuentra cada una de ellas en función de la probabilidad de cada recomendador. Recordamos que al definir el modelo y explicar el descubrimiento en la sección 3.2.2 se planteó la idea de representar todas las fuentes de descubrimiento como recomendadores y asignar a cada uno de ellos una probabilidad que determina la frecuencia con la que un usuario utiliza ese recomendador para encontrar una película. Este diseño simplifica la implementación, pues no es necesario estar considerando cada posible fuente de descubrimiento (búsqueda, publicidad, descubrimiento aleatorio…) por separado.

Se han implementado cuatro recomendadores distintos, pero el programa permite introducir tantos como se desee:

- Recomendador aleatorio uniforme: elige la película al azar siguiendo una distribución uniforme, es decir, ningún película tiene más probabilidad de ser recomendada que otra.
- Recomendador aleatorio sesgado: recomienda unas películas con más probabilidad que otras, siguiendo una distribución arbitraria para cada ítem. En los experimentos realizados, se ha utilizado una distribución power law de estructura similar a la que sigue la relevancia, pero generada independientemente de ésta. Este recomendador

permite, por ejemplo, asemejarse a una distribución de descubrimiento por publicidad, donde la probabilidad de encontrar un ítem depende de la inversión en publicidad (i.e. difusión) que haya tenido el mismo.

- Recomendador por popularidad: recomienda los ítems por orden decreciente del número de ratings que poseen hasta el momento.
- Recomendador social: recomienda los ítems por orden decreciente del número de ratings relevantes entre los vecinos del usuario en la red social.

En la simulación, sin embargo, sólo están incluidos el recomendador uniforme y el sesgado y sus probabilidades son parámetros del sistema: *Random Probability* y *Biased Random Probability*. En el caso del recomendador sesgado también es necesario indicar el parámetro α de la función power law con la que se simula (*Marketing alpha*).

4.2.2 Consumición

Recordemos de la explicación de la consumición al definir el modelo sección 3.2.3 que, por simplicidad, el número de películas que un usuario consume en un turno se modeliza mediante un número fijo. Este número es un parámetro de la simulación (*Nr. watched movies per step*).

Dichas películas se eligen de una en una y de manera uniforme del conjunto de películas descubiertas pero todavía no consumidas por el usuario. Puede ocurrir que el usuario no haya descubierto suficientes películas, en cuyo caso se consumen todas las que hay. Cuando una película es seleccionada, se extrae de dicho conjunto y se marca cómo vista.

4.2.3 Votación

Tras consumir una película, y antes de seleccionar la siguiente, se decide si se vota o no. Para ello, se tienen en cuenta los parámetros $p(rate|R)$ y $p(rate|\neg R)$ que representan, respectivamente, la probabilidad de votar dado que la película es relevante y dado que no lo es. Como ya se dijo en la sección 3.1, se ha observado que, en general, los votos de los usuarios están sesgados hacia los casos positivos, es decir, que $p(rate|R) > p(rate|\neg R)$.

Si se realiza la votación, se asigna a la película un valor 1 si es relevante para el usuario y un valor 0 si no lo es.

4.2.4 Comunicación: flujo en la red social.

La propagación de información en la red social se basa en el proceso de comunicación llevado a cabo entre los usuarios. Se consideran cuatro factores que determinan la forma de llevar a cabo esta comunicación:

- Momento en el que se desarrolla la conversación. De acuerdo con este factor se diferencian dos modos: modo síncrono y modo asíncrono.

  En el modo síncrono la comunicación se realiza en el mismo proceso que la consumición, esto es, el usuario inicia la conversación tras consumir una película.

  En el modo asíncrono, el usuario espera al final de su turno, cuando ya ha finalizado todas las interacciones anteriores (descubrimiento, consumición y votación) para comunicarse con sus vecinos.

- Dirección del flujo de información: los usuarios pueden participar en la conversación sólo como emisores, siendo los únicos que transmiten información al

conversar con un amigo, o también como receptores, esto es, preguntando al amigo y recibiendo la información que éste les transmita.

- Número de películas de las que habla un usuario al transmitir información a un amigo. Por simplicidad se consideran sólo dos opciones: hablar de sólo una película o hablar de todas las películas que conoce dicho usuario.
- Número de amigos con los que entablar conversación en cada turno. Existen varias variantes a este respecto, que se explicarán a continuación, pero se resumen en dos: comunicarse con un único amigo o comunicarse con varios.

La elección entre las distintas opciones que atañen a cada factor se realiza mediante parámetros booleanos que se explican a continuación, junto con un análisis más en detalle de cada punto.

4.2.4.1 Modo síncrono vs. asíncrono

Un usuario puede decidir iniciar una conversación justo después de ver una película, o en cualquier otro momento. Si decide hacerlo después de la consumición, lo lógico es que la película acerca de la que informe sea la que acaba de ver. En cambio, si lo realiza en cualquier otro momento, ninguna película tiene preferencia de ser el tema de conversación.

Este matiz en el momento de inicio de la conversación nos lleva a diferenciar entre dos modos de comunicación: modo síncrono y modo asíncrono. La elección entre uno u otro se realiza mediante el parámetro *Synchronous mode*, que toma un valor verdadero para el modo síncrono y un valor falso para el modo asíncrono.

En el modo síncrono, los usuarios deciden acerca de contar una película a sus amigos únicamente después de verla. En la realidad esto no ocurre de forma tan estricta, pero en general las conversaciones sobre la mayoría de las películas que vemos tienen mayor frecuencia en un entorno de tiempo cercano a haberlas visto. Las películas de las que hablamos tiempo más tarde suelen ser muy pocas en relación con la totalidad de las que vemos. Pese a ello, esta última posibilidad se incluye en el modo asíncrono.

Como consecuencia de esta restricción en el modo síncrono, los usuarios establecen como mucho una conversación por iniciativa propia por cada película consumida. Además, si deciden no hablar de ella, ya no podrán iniciar una conversación para comunicar dicha película, pues no podrán volver a consumirla. Pese a ello, todavía podrán contar que la han visto si son preguntados por algún amigo.

En el modo asíncrono, el proceso de comunicación es independiente del de consumición y los usuarios deciden comunicarse con sus amigos en un momento diferente a cuando han visto las películas. De hecho, pueden elegir cualquier película/s para transmitir.

Otro aspecto del modo asíncrono es que necesitamos controlar el número de usuarios que inician una conversación para que la comunicación por la red social no aumente hasta valores inverosímiles. Para ello incorporamos un parámetro $p(start)$ que determina la probabilidad a priori de que un usuario inicie el proceso de comunicación en el modo asíncrono al acabar su turno. También se puede interpretar este parámetro como el ratio de usuarios que "hablan" en cada unidad de tiempo.

El motivo por el que este control no es necesario en el modo síncrono es porque se lleva a cabo al mismo ritmo que el proceso de consumición, el cual ya está controlado por el descubrimiento y el número fijo de películas que se consumen. Además, en este

modo sólo se puede hablar de la película que se acaba de consumir, mientras que en el modo asíncrono podríamos hablar de todas las películas consumidas, disparando así el descubrimiento.

El parámetro $p(start)$ no anula la influencia de la relevancia, es decir, si el usuario ha decidido incorporarse al proceso de comunicación en el turno actual, para cada película de la que considere hablar todavía deberá decidir si lo hace o no vía los parámetros $p(tell|R)$ y $p(tell|\neg R)$.

4.2.4.2 Contar vs. contar y preguntar

En una conversación tradicional entre dos usuarios, la información fluye en ambos sentidos, es decir, ambos toman "turnos" para hablar y cuando el otro habla, escuchan – cuando menos en una conversación ideal. Sin embargo, una comunicación en la que sólo uno de los usuarios transmite y el otro simplemente recibe, también es posible, incluso se puede dar en la realidad. Un ejemplo de este tipo de comunicación es la que se produce en la red social Twitter: cuando un usuario escribe un tweet está trasmitiendo información a sus seguidores pero no tiene por qué fluir información en sentido contrario y, de hecho, si el seguimiento no es recíproco, los seguidores no pueden enviar información de vuelta a este usuario. Caso semejante es cualquier sistema de emisión de información en modo broadcast (radio, TV, prensa tradicional, etc.)

En función de quién tome la iniciativa en el flujo de la información dentro de la conversación, podemos diferenciar dos casos: comunicación hacía adelante (contar) y comunicación hacía atrás (preguntar).

En la comunicación hacia delante el usuario cuenta a sus amigos las películas que ha visto, por iniciativa propia. En la comunicación hacía atrás el usuario pregunta por la experiencia de sus contactos, y éstos se la transmiten. En los modelos de comunicación que contemplamos en nuestro caso, la comunicación hacia atrás tiene lugar siempre después de haberse producido la comunicación hacía delante, es decir, un usuario cuenta siempre una experiencia suya antes de preguntar por la del vecino. Esta decisión es simplemente por convención, pues el orden en que intervengan los usuarios no es realmente importante; se trata de dar lugar a una conversación bidireccional y un mensaje en cada sentido es la formulación más simple.

Cuando un usuario es preguntado por un amigo, elige una/s película/s que haya visto para transmitir a dicho amigo.

En ambos tipos de comunicaciones, siempre que un usuario vaya a transmitir una película, se tienen en cuenta las probabilidades $p(tell|R)$ y $p(tell|\neg R)$. Esto incluye al usuario que es preguntado y ya ha elegido la/s película/s para contestar.

En la simulación, la elección entre una forma de comunicar y otra se determina con el parámetro *ask* que toma un valor verdadero si se realiza comunicación hacía atrás, es decir, si se pregunta, y un valor falso si sólo se cuenta, esto es, si sólo existe comunicación hacia delante.

4.2.4.3 Una película vs. todas las películas

Transmitir información de (potencialmente) todas las películas, en vez de sólo de una, es siempre una opción en la comunicación hacía atrás, es decir, cuando un usuario debe contestar a otro puede hablarle de todas las películas que ha visto. Sin embargo, en la comunicación hacía delante sólo en el modo asíncrono es posible informar sobre todas las películas que ha visto un usuario, porque en el modo síncrono sólo se informa sobre la película que se acaba de consumir.

La elección entre hablar acerca de todas las películas o sólo de una está determinada por el parámetro *Tell about all movies*. Por lo que acabamos de indicar, sólo se admite un valor verdadero en este parámetro si se ha establecido un modo asíncrono de comunicación.

Si sólo se habla de una película, esta película se elige uniformemente entre el conjunto de películas vistas por el usuario. Si se habla de todas, cada película tiene su oportunidad de ser transmitida. En ambos casos, se debe pasar el filtro final determinado por las probabilidades $p(tell|R)$ y $p(tell|\neg R)$.

4.2.4.4 Único amigo vs. todos los amigos

Este parámetro establece el número de amigos con los que contacta un usuario para transmitir información acerca de la/s película/s que ha elegido. Por ello, sólo afecta a la comunicación hacía delante, pues en la comunicación hacía atrás el usuario sólo contesta al usuario que le ha preguntado.

Existen dos posibilidades, hablar con todos los amigos o hacerlo sólo con uno. La elección entre ambas se realiza mediante el parámetro *Tell to all friends* que tiene un valor verdadero si se comunica a todos los amigos y un valor falso si sólo se habla con uno. A continuación se explica cada una de estas opciones.

- Único amigo: en este caso, el usuario sólo contacta con un amigo para transmitirle información. Este funcionamiento es el mismo tanto para el modo síncrono como para el asíncrono.

  A la hora de seleccionar al amigo se tienen en cuenta los que han sido seleccionados en los turnos anteriores para no volver a escogerlos. Respecto a qué amigo elegir entre los restantes, se consideran dos posibilidades: elegirlo al azar con probabilidad uniforme o seleccionarlo de forma secuencial, es decir, cada usuario tiene una lista ordenada de amigos y en cada turno se elige el siguiente al que se había elegido en el turno anterior. En ambos casos cuando ya se han elegido todos los amigos, se reinicia el recorrido. El parámetro *Choose a random neighbour* determina cuál de las dos opciones se lleva a cabo tomando un valor verdadero para la selección aleatoria y un valor falso para la selección secuencial.

- Todos los amigos: la comunicación hacía adelante considera todos los contactos del usuario, pero de forma distinta en el modo síncrono que en el asíncrono.

  En el modo síncrono, la/s película/s son comunicadas a todos los amigos uno por uno, aplicando cada vez el filtro $p(tell|R)$ y $p(tell|\neg R)$.

  En el modo asíncrono, la/s película/s son comunicadas a un ratio $p(go)$ de todos los amigos, es decir, $p(go)$ determina la probabilidad de que dado un amigo, el usuario lo selecciona para conversar. Para cada uno de los seleccionados, $p(tell|R)$ y $p(tell|\neg R)$ determinan si definitivamente se trasmite o no la información de la película.

  La razón por la que se introduce este parámetro es la misma por la que introducíamos el parámetro $p(start)$, para controlar un posible flujo desproporcionado de información en comparación con el modo síncrono.

4.2.4.5 Resumen

En resumen, la comunicación se desarrolla en las siguientes etapas.

1. El usuario que toma la iniciativa selecciona uno o varios amigos para transmitirles información. Para cada uno de ellos realiza los pasos que siguen.
2. En el modo asíncrono, el usuario elige una o varias películas para comunicar. En el modo síncrono, la película ya está determinada (es la que se acaba de consumir).
3. Dependiendo de las probabilidades $p(tell|R)$ y $p(tell|\neg R)$, el usuario transmite en su entorno social la información acerca de cada película.
4. Si está activada la comunicación hacía atrás y realmente se ha transmitido la información, a la vez que habla de una película a un amigo, el usuario pregunta al interlocutor por la/s película/s que éste ha visto. Dicho amigo ejecuta los pasos 2 y 3 para contestarle.

La Tabla 4 resume las posibles combinaciones de parámetros y en qué momento intervienen.

	Modo síncrono		**Modo asíncrono**	
	Contar	**Preguntar**	**Contar**	**Preguntar**
Usuario que transmite	Usuario que acaba de ver la película	Amigo preguntado	Ratio de usuarios determinado por *p(start)*.	Amigo preguntado
Película que se transmite	La película que el usuario acaba de consumir	Dos opciones (determinadas por el parámetro *Tell about all movies*): a) Iterar sobre todas las películas b) Elegir uniformemente una entre todas las películas vistas.		
Seleccionar un amigo	Dos opciones (determinadas por el parámetro *Tell to all friends)*: a) Iterar sobre todos los amigos b) Elegir un amigo: dos opciones (parámetro *Choose a random neighbour*). b.1) de forma uniforme b.2) de forma secuencial	El usuario que ha realizado la pregunta	Dos opciones (determinadas por el parámetro *Tell to all friends*): a) Iterar sobre un ratio de amigos determinado por *p(go)*. b) Elegir un amigo: dos opciones (parámetro *Choose a random neighbor*). b.1) de forma uniforme b.2) de forma secuencial	El usuario que ha realizado la pregunta
Contar acerca de la película al amigo	Tomar una decisión basada en la relevancia y determinada por los parámetros $p(tell\|R)$ y $p(tell\|\neg R)$			

Tabla 4. Resumen de los distintos modos de comunicación y los parámetros que los determinan.

4.3 Condiciones de parada de la simulación

Se consideran cuatro posibles condiciones de parada para finalizar la simulación:

- Timeout: la simulación finaliza cuando se alcanza un cierto número de iteraciones (unidades de tiempo).
- Número de ratings: la simulación se detiene cuando se acumula un cierto número de ratings producidos por los usuarios.
- Todas las películas descubiertas: la simulación acaba cuando todas las películas han sido descubiertas por todos los usuarios.
- Todas las películas vistas: la simulación finaliza cuando todas las películas han sido vistas por todos los usuarios. Este punto casi nunca se alcanza, a no ser que $p(tell|R)$ y $p(tell|\neg R)$ valgan ambos 1, pues si se decide no ver una película, ya no se puede volver a considerar, y esa película queda sin ser vista por ese usuario.

Para elegir entre una condición de parada u otra se introducen cuatro parámetros booleanos que representan cada una de las anteriores condiciones y que tienen un valor verdadero si esa condición está vigente y un valor falso en caso contrario. Estos parámetros son, respectivamente, *Use timeout, Fixed nr. of ratings, All movies found by all users* y *All movies watched by all users*.

Para el caso de parada por timeout o número de ratings, se añaden dos nuevos parámetros que representan los valores concretos de parada, *timeout* y *nr. of ratings*.

4.4 Valores de salida de la simulación

Los datos que genera la simulación y que nos interesa observar son, para cada ítem, los ratios de usuarios que han interaccionado de determinada forma con él. En función del tipo de interacción tenemos los siguientes ratios:

- $p(found|i) = p(f|i)$: ratio de usuarios que han encontrado el ítem i.
- $p(\text{watch}|i) = p(w|i)$: ratio de usuarios que han visto el ítem i.
- $p(rate|i) = p(r|i)$: ratio de usuarios que han votado el ítem i.
- $p(rate, relevant|i) = p(r, R|i)$: ratio de usuarios para los cuales era relevante el ítem i y lo han votado.
- $p(rate, not\ relevant|i) = p(r, \neg R|i)$: ratio de usuarios para los cuales no era relevante el ítem i y lo han votado.
- $p(relevant|i) = p(R|i)$: ratio de usuarios para los cuales es relevante el ítem i.

De ahora en adelante, cuando nos refiramos a estos ratios omitiremos la condición al ítem i, es decir, hablaremos de $p(f), p(w)$… Esta notación, añade simplicidad y permite hacer referencia a la función que devuelve, para cada ítem, el correspondiente ratio de usuarios que lo descubren/consumen/votan…

Si promediamos sobre los ítems, también podemos observar el ratio de usuarios que, de media, realizan un descubrimiento, votación, consumición…

A parte de los anteriores valores que se pueden considerar prioritarios, podemos observar otras relaciones de interés:

- Promedio del grado inicial: en el caso en que el descubrimiento por fuentes externas sea nulo (parámetro *Found / watched ratio* a 0) puede ser interesante considerar el grado de los usuarios que conocen inicialmente la información sobre las películas para poder relacionarlo con la velocidad de propagación.
- Iteraciones y ratings realizados al finalizar el programa, pues otorgan una cierta medida de la velocidad a la que se ha propagado la información.

4.4.1 Ajuste

Es parte de los objetivos del trabajo que el modelo permita explicar cómo se generan distribuciones como las que se observan en datasets reales. Así, se puede configurar la simulación para que, además de generar los valores de salida anteriormente explicados, mida la similitud entre los datos simulados y los datos de algún dataset real, por ejemplo MovieLens.

En concreto, nos interesa contrastar las distribuciones de ratings, tanto positivos como negativos, generadas por la simulación – curvas $p(rate)$, $p(rate, relevant)$ y $p(rate, not\ relevant)$ – con las correspondientes curvas de datos reales. Para ello, se ha dotado a la simulación de la opción de cargar un fichero con ratings reales, de forma que en la aplicación se visualice la generación de datos reales en el tiempo (usando las marcas temporales, esto es, el instante de tiempo en que el usuario correspondiente asignó el rating al ítem, que generalmente están disponibles en los datasets) junto a la generación de datos por el modelo simulado.

Si se elige esta opción de cargar datos de ratings reales desde un fichero, se asume que la condición de parada es por número de ratings, es decir la simulación se detiene cuando se generan tantos ratings como contiene el fichero. También se asigna automáticamente al parámetro *Nr. of items* (número de películas) el valor del número total de ítems que aparecen votados en el dataset.

El hecho de que en la aplicación se muestre la evolución en el tiempo de las curvas de todos los valores de salida, incluidas las del conjunto de datos reales, permite observar cómo se parecen las curvas de ratings – $p(rate)$, $p(rate, relevant)$ y $p(rate, not\ relevant)$ – simuladas a las reales. Sin embargo, también se ha llevado a cabo un análisis más teórico de la similitud entre ambas curvas. Existen diferentes medidas y principios para comparar cuánto se parecen dos distribuciones o series de datos que suelen emplearse con diferentes fines tales como, en particular para el caso que nos ocupa, medir qué tal ajusta una curva de distribución teórica (modelo) a una observación empírica (muestra). En nuestro caso utilizamos para ello las siguientes medidas que suelen ser habituales en el ámbito en el que nos movemos.

- KLD (Kullback–Leibler divergence): es una medida de la diferencia entre dos distribuciones de probabilidad, P y Q, que viene expresada por la siguiente fórmula:

$$D_{KL}\,(P\|Q) = \sum_{i} \ln\left(\frac{P(i)}{Q(i)}\right) P(i)$$

  Cabe destacar que esta medida es asimétrica, es decir:

$$\mathrm{D_{KL}}\,(\mathrm{P}\|\mathrm{Q}) \neq \mathrm{D_{KL}}\,(\mathrm{Q}\|\mathrm{P})$$

  En nuestro caso, consideramos P la distribución de los datos reales y Q la obtenida por la simulación. Ambas se evalúan sobre los distintos ítems, por lo que la variable i del sumatorio recorre todos los ítems.

- Error medio: consiste simplemente en calcular la media de las diferencias (errores) entre ambas distribuciones.

$$\frac{1}{\mathcal{I}}\sum_{i}|P(i)-Q(i)|$$

  Donde $\mathcal{I}$ es el número de ítems.

- Error medio relativo: en este caso se calcula la media de las diferencias relativas.

$$\frac{1}{\mathcal{I}}\sum_{\mathrm{i}}\frac{|\mathrm{P(i)}-\mathrm{Q(i)}|}{\mathrm{P(i)}}$$

  Al relativizar el error medio se consigue ver su proporción respecto a los valores de las distribuciones P y Q. Así, un mismo error medio da lugar a un error relativo menor si las distribuciones toman valores grandes que si los toman pequeños.

También se ha sometido la similitud entre las curvas reales y simuladas a dos test de bondad de ajuste: el test chi-cuadrado de Pearson (Greenwood 1996) y una variación del test de Kolmogorov–Smirnov para distribuciones discretas (Arnold 2011) que se encuentra implementada en el lenguaje de programación R. Para probar este último se ha utilizado la librería JRI[7] que permite ejecutar código R desde Java. Sin embargo, estos test son demasiado estrictos y curvas que la interfaz muestra prácticamente idénticas no son aceptadas por ninguno de los test. De hecho rara vez en la literatura de ajuste de datos en campos relacionados con la recuperación de información o la recomendación en general se encuentran modelos que superan estos tests. Por ello, y como el objetivo de medir la diferencia entre ambas curvas es encontrar la configuración de parámetros que mejor ajusta la curva real y no es un requisito pasar ningún test, se ha decido utilizar únicamente las medidas de similitud (KLD y errores medios).

Una última cuestión que hace referencia al ajuste de datos reales es que, al igual que se puede observar la evolución en el tiempo de dichos datos, también es posible observar directamente los valores finales, es decir, el ratio total de usuarios que ha votado un cierto ítem. Ambas consideraciones resultan interesantes, pues la primera permite comparar datos reales y simulados en cada instante de tiempo, y la segunda observar cómo la simulación se acerca o aleja en el tiempo de los datos finales a los que debería igualar. La visualización de estas observaciones y mediciones se describe en el apartado 5.3.

[7] http://rforge.net/JRI

4.5 Tabla de parámetros

En la Tabla 5 se muestran todos los parámetros de la simulación, que se han ido explicando en los apartados anteriores.

Parámetro	Descripción
Nr.of Items	Número de ítems
Graph	Tipo de grafo que representa la red social
Nr. of Users	Número de usuarios
Average degree	Grado promedio del grafo que representa la red social
$p(rel)$	Priori de relevancia
Relevance alpha	Parámetro de la distribución de relevancia.
Found / watched ratio	Ratio entre películas descubiertas y vistas durante el turno de un usuario
Initial Nr Users	Número de usuarios que conocen las películas desde el principio. Sólo se considera si r es 0.
Nr. watched movies per step	Número de películas que se consumen en cada turno de un usuario.
$p(rate\|R)$	Probabilidad de votar un ítem dado que es relevante para el usuario.
$p(rate\|\neg R)$	Probabilidad de votar un ítem dado que no es relevante para el usuario.
$p(tell\|R)$	Probabilidad de hablar de un ítem dado que es relevante para el usuario.
$p(tell\|\neg R)$	Probabilidad de hablar de un ítem dado que no es relevante para el usuario.
ask	Parámetro booelano que indica si existe comunicación hacia atrás (*true*) o si no (*false*).
Synchronous mode	Parámetro booleano que indica si la comunicación se lleva a cabo de forma síncrona (*true*) o no (*false*).
Tell to all fiends	Parámetro booleano que indica si los usuarios cuentan una película a todos sus amigos (*true*) o sólo a uno (*false*).
Tell about all movies	Parámetro booleano que indica si los usuarios hablan de todas las películas consumidas (*true*) o no (*false*). Solo se acepta un valor verdadero en el modo asíncrono.
Choose a random neighbor	Si sólo se elige a un amigo para hablar, este parámetro booleano índica si es el siguiente al que se había elegido anteriormente (*false*) o se elige al azar (*true*).

$p(go)$	Ratio de amigos con los que habla un usuario en el modo asíncrono si *Tell to All fiends* es true.
$p(start)$	Probabilidad a priori de que un usuario inicie el proceso de comunicación en el modo asíncrono.
Fraction of users per time step	Indica la fracción del número de usuarios que se recorren por unidad de tiempo.
Random probability	Probabilidad de que el recomendador uniforme sea elegido como fuente de descubrimiento.
Biased Random Probability	Probabilidad de que el recomendador sesgado (publicidad) sea elegido como fuente de descubrimiento.
Marketing alpha	Parámetro α de la distribución power law utilizada por el recomendador sesgado que simula la publicidad.
All movies found by all users	Parámetro booleano que determina si la simulación finaliza cuando todos los usuarios han descubierto todas las películas (*true*) o no (*false*).
All movies watched by all users	Parámetro booleano que determina si la simulación finaliza cuando todos los usuarios han visto todas las películas (*true*) o no (*false*).
Fixed nr. of ratings	Parámetro booleano que determina si la simulación finaliza cuando se alcanza un cierto número de ratings (*true*) o no (*false*).
Nr. of ratings	Número de ratings en el cual se detiene la simulación si está activada la condición de parada por número de ratings.
Use timeout	Parámetro booleano que determina si la simulación finaliza cuando se alcanza un cierto número de iteraciones (*true*) o no (*false*).
timeout	Número de iteraciones tras las cuales finaliza la simulación si la condición de parada es por timeout.

Tabla 5. Lista de parámetros de la simulación

5. Detalles de implementación

En los apartados anteriores se ha definido el modelo a desarrollar así como los patrones de comportamiento que permiten simular dicho modelo. En este apartado se describen los detalles implementativos de la simulación, tales como los módulos en los que se divide el programa, el diseño de la base de datos o la interfaz de usuario.

5.1 Módulos

El sistema desarrollado en este TFG consta de cinco módulos principales, que se explican brevemente a continuación.

5.1.1 Proceso central de la simulación

Este módulo engloba todos los procedimientos que intervienen en el bucle principal (descubrimiento, consumición, votación, modos de comunicación, condiciones de parada…), así como las variables necesarias para llevarlos a cabo. Entre estas variables destacan:

- Lista de películas relevantes para cada usuario.
- Lista de películas descubiertas y no consumidas por cada usuario.
- Lista de películas consumidas por cada usuario.
- Ratio de usuarios que ha descubierto/consumido/votado cada ítem. En el caso de que se estén ajustando datos reales también se incluyen los ratios reales de ratings.
- Promedios de los ratios anteriores.

Estas variables y los métodos que trabajan con ellas se encuentran distribuidos en dos clases, RatingModel y NetworkRatingModel. La clase RatingModel representa un modelo básico de generación de votos y contiene la información referente al número de usuarios y de ítems junto con las listas que indican qué usuario ha votado qué ítem y con qué voto. En esta clase no se consideran aspectos como la relevancia o la red social. La clase NetworkRatingsModel, que extiende de la anterior, añade los elementos característicos del modelo definido en este trabajo. Así, contiene la red social, la distribución de relevancia y los distintos métodos que se encargan de emular el comportamiento por turnos de los usuarios: descubrimiento, consumición, comunicación y votación, este último heredado de la clase padre.

5.1.2 Grafos

El módulo de grafos se encarga de la generación y gestión del grafo que representa la red social.

En la simulación, la red social interviene únicamente a la hora de obtener los vecinos de cada usuario por lo que las estructuras que se implementan para representar los grafos sólo es necesario que sean capaces de satisfacer este requisito. Así, se ha definido la interfaz *GenericGraph*, que deben implementar dichas estructuras y que requiere los siguientes tres métodos:

- getNeigborsIterator (node): que devuelva todos los vecinos de un nodo.

- getRandomNeigbor (node): que devuelva un vecino al azar.
- getNextNeigbor (node): que devuelva el siguiente vecino según una iteración secuencial de orden fijo (pero arbitrario).

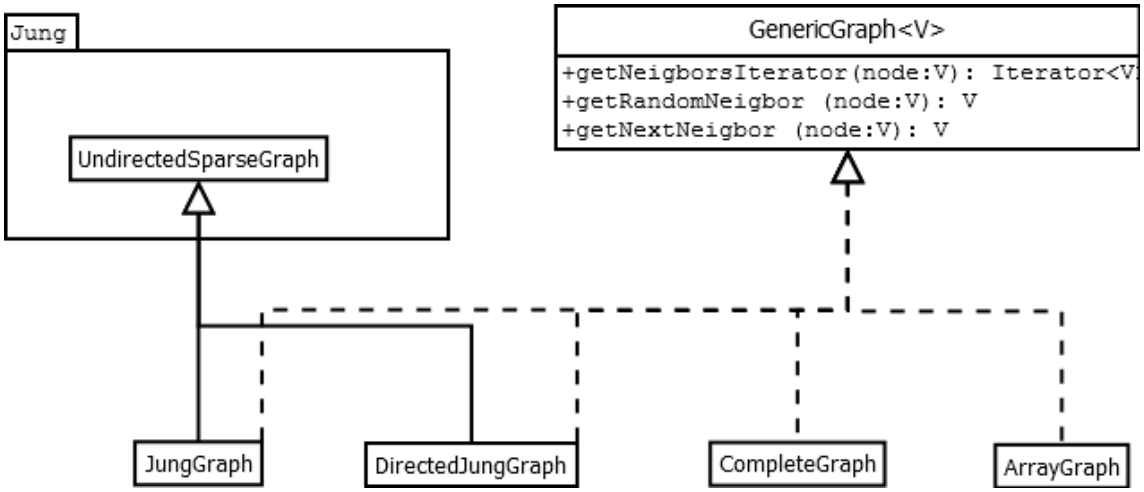


Figura 9. Diagrama de clases del módulo de grafos.

En la Figura 9 se puede observar el diseño de clases del módulo de grafos. Así, vemos que se han desarrollado cuatro clases que implementan la interfaz *GenericGraph*, a saber, *JungGraph, DirectedJungGraph, CompleteGraph* y *ArrayGraph*. En función de los requisitos del grafo que se quiere utilizar como red social, conviene utilizar una implementación u otra, tal y como se explica a continuación.

5.1.2.1 Jung : clases JungGraph y DirectedJungGraph

Para grafos de pequeño tamaño se ha utilizado la libreía Jung de Java, que incluye métodos de generación y gestión de grafos. Para generar grafos de mayor escala hemos utilizado la librería SNAP, implementada en C++ y más eficiente. Esta librería la utilizamos externamente, es decir, volcando la salida del programa a un archivo que después se toma como entrada de nuestro programa.

La generación de grafos mediante la librería Jung se lleva a cabo en la clase GraphBuilder, a la que se llama directamente desde el programa, y que incluye métodos para generar grafos aleatorios (no sólo los dos tipos con los que hemos experimentado – Barabási y Erdös – sino otros cuantos grafos aleatorios comunes) así como funciones para cargar grafos externos desde un archivo con formato csv o gdf.

Para dar soporte y recorrido a los grafos leídos o generados por la clase GraphBuilder se han desarrollado dos implementaciones de la interfaz GenericGraph, JungGraph y DirectedJungGraph, que extienden del grafo de Jung y representan, respectivamente, los grafos no dirigidos y dirigidos. Estas clases son las que se utilizan por defecto en el programa y con las que se han gestionado todos los grafos a excepción de los grafos (Twitter y Orkut) que por su gran escala han precisado una implementación propia optimizada al efecto que explicamos en el punto 5.1.2.3.

5.1.2.2 Clase CompleteGraph

Aunque también se podía haber utilizado Jung para representar grafos completos, esto supone un coste en memoria innecesario. Para representar un grafo completo, los dos primeros métodos del API descrita por GenericGraph no precisan de ningún tipo de estructura, porque en un grafo completo los vecinos de un usuario son el resto de

usuarios. Para el último método consideramos una estructura de array en la que, para cada usuario, se indica el siguiente vecino a devolver según el recorrido secuencial, tal y como se muestra en la Figura 10.

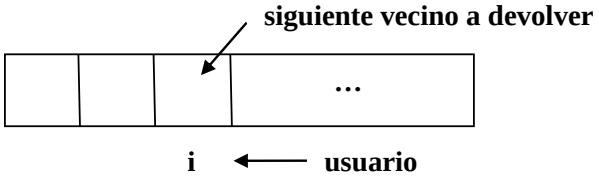


Figura 10. Diseño de la estructura interna del grafo completo.

Para almacenar un grafo, Jung utiliza una estructura de HashMap que indica, para cada vértice, sus vecinos. Además, también posee otro HashMap para las aristas, que indica los nodos inicial y final de cada una de ellas. Esto implica que si el grafo tiene N nodos y es completo – cada nodo tiene $N-1$ vecinos – Jung supone un coste en memoria de $O\big(N \times (N-1)\big) = O(N^2)$ para almacenar los vértices y $O(E) = O\left(\frac{N\times(N-1)}{2}\right) = O(N^2)$ para la aristas.

Al utilizar nuestra estructura de grafo completo reducimos, por tanto, la memoria empleada por Jung, $O(N^2)$, a un solo array, $O(N)$. El tiempo de acceso, sin embargo, sigue siendo $O(1)$ en ambos casos.

5.1.2.3 Clase ArrayGraph.

Como acabamos de ver, la implementación de grafos en Jung optimiza el tiempo de acceso sobre el coste de memoria, por lo que presenta limitaciones para manejar redes sociales del orden de millones de usuarios. Por ello, para este tipo de situaciones hemos diseñado una estructura de grafo que optimiza el coste en memoria. A la clase que presta esta funcionalidad se la ha denominado ArrayGraph, por representar el grafo mediante dos arrays.

- El primer array indica, para cada usuario, la posición del siguiente array en la que se encuentran sus vecinos.
- El segundo array contiene los vecinos de cada usuario, colocando los de uno a continuación de los de otro.

  En la Figura 11 podemos observar este esquema:

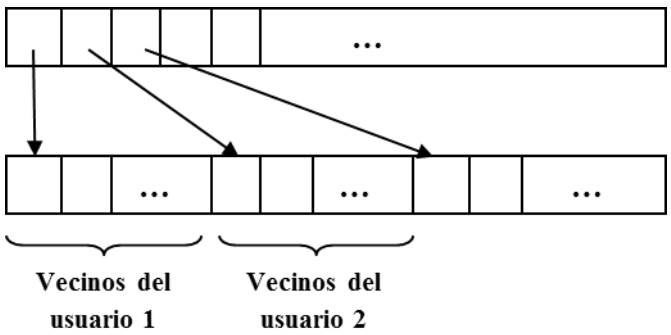


Figura 11. Diseño de la estructura interna de ArrayGraph.

Veamos la optimización que obtenemos con esta estructura de grafo. Consideramos un grafo con N nodos y E aristas. En el HashMap de vértices que utiliza Jung se están

almacenando todas las aristas, pues cada usuario y cada vecino de su lista de vecinos forman una arista, luego en total ambos HashMap están utilizando $2 \times E$ memoria. En este caso vamos a evitar la notación *O grande*, porque nos interesan las constantes dado que el doble de memoria cuando se están utilizando 4 GB puede ser la diferencia entre poder o no ejecutar la aplicación. Además, la estructura HashMap nos asegura que, de media, un 75% de la memoria reservada está siendo ocupada, lo cual supone que en promedio se está reservando un tercio más de lo que se utiliza.

En nuestra estructura, por el contrario, se necesita N para el primer array y E para el segundo. Si añadimos el hecho de que la mayoría de grafos con los que estamos tratando poseen muchas más aristas que nodos –el grafo de Twitter posee 40 veces más aristas que nodos – podríamos considerar el coste del primer array despreciable frente al segundo. Así, tendríamos que un grafo de Jung necesita más del doble de memoria que la estructura ArrayGraph.

5.1.3 Recomendadores

En este módulo se implementan los recomendadores explicados anteriormente en el apartado 4.2.1.

Para ello hemos diseñado una clase raíz denominada Recommender que representa un algoritmo general de recomendación. Contiene como atributo la clase en la que se lleva a cabo el proceso principal (NetworkRatingModel) y en la que se encuentran las variables que informan del estado de la simulación en las cuales suelen basarse los recomendadores. Dispone además del método abstracto *recommendTopN (int user, int n)* que devuelve las n películas que el algoritmo recomienda al usuario indicado.

Este diseño permite crear en cualquier momento un nuevo recomendador e incluirlo en la simulación. Basta con hacerlo extender de la clase Recommender e incluirlo en la lista de recomendadores de los que la simulación hace uso a la hora de determinar lo que descubren los usuarios.

5.1.4 Estadísticos

Este módulo contiene las implementaciones de las medidas de similitud y los test de bondad de ajuste descritos en el apartado 4.4.1, a saber, la distancia KLD, el error medio, el error medio relativo, el test chi-cuadrado de Pearson (Greenwood 1996) y el test de Kolmogorov–Smirnov para distribuciones discretas (Arnold 2011).

El módulo consta únicamente de una clase estática en la que se implementan los métodos anteriores. Todos ellos reciben como parámetros los valores de las dos curvas entre las que medir la similitud y devuelven dicha similitud como un valor decimal.

5.1.5 Interfaz de usuario

En el módulo de interfaz se implementan las distintas ventanas de la vista. Una explicación más detallada de dicha interfaz y de las ventanas que la componen se aporta en el apartado 5.3.

Únicamente cabe destacar que la implementación de las distintas ventanas se ha llevado a cabo haciendo uso de la librería Swing. Respecto a los puntos de las curvas que se muestran en la interfaz se han dibujado pixel a pixel, calculando las coordenadas que debería ocupar cada punto sin usar ninguna librería alternativa.

5.2 Base de datos

El programa se integra con una base de datos MySQL en la que se escriben tanto los parámetros como los resultados de la simulación. El motivo por el que se guardan estos valores es para poder relacionarlos posteriormente, seleccionando diferentes vistas, cortes y rangos de variables y dependencias a observar en gráficas y tablas de cara al análisis de resultados. El almacenamiento en base de datos permite además sincronizar datos con vistas en Excel de cara a generar tablas dinámicas y gráficas, como algunas que se mostrarán en la sección 6.4.

El diagrama ER de la base de datos se muestra en la Figura 12.

Las clases *Run* (Ejecución), *InPut* (Entrada) y *Output* (salida) representan, respectivamente, la configuración de parámetros de una ejecución, la ejecución en sí, y los valores de salida que produce dicha ejecución. En la entidad *Output* el atributo *Sum of p(distribución)* se refiere a la integral del promedio de la distribución correspondiente a lo largo del tiempo.

Respecto a la entidad *Point*, representa un punto de valores promedio por ítem en el tiempo, es decir, los valores de los promedios de $p(f)$, $p(w)$, $p(r)$, $p(r|R)$ y $p(r|\neg R)$ en un cierto instante de tiempo, determinado por el atributo *time*. El atributo *Type* indica el tipo de dicho instante, y puede ser: *timeout* (momento final de la simulación), *timeout/2* (momento en que la simulación alcanza la mitad de las iteraciones planificadas), *ratings* (momento en que se supera el número de ratings indicado como argumento), *all movies found* (momento en que todos los usuarios han encontrado todas las películas) o *all movies watched* (momento en que todos los usuarios han visto todas las películas). Es decir, aparte de las integrales, tenemos en cuenta los valores de las curvas de promedios en las condiciones de parada. Además, en el caso de parada por timeout, también consideramos los valores cuando ha pasado la mitad del tiempo, como punto de control intermedio.

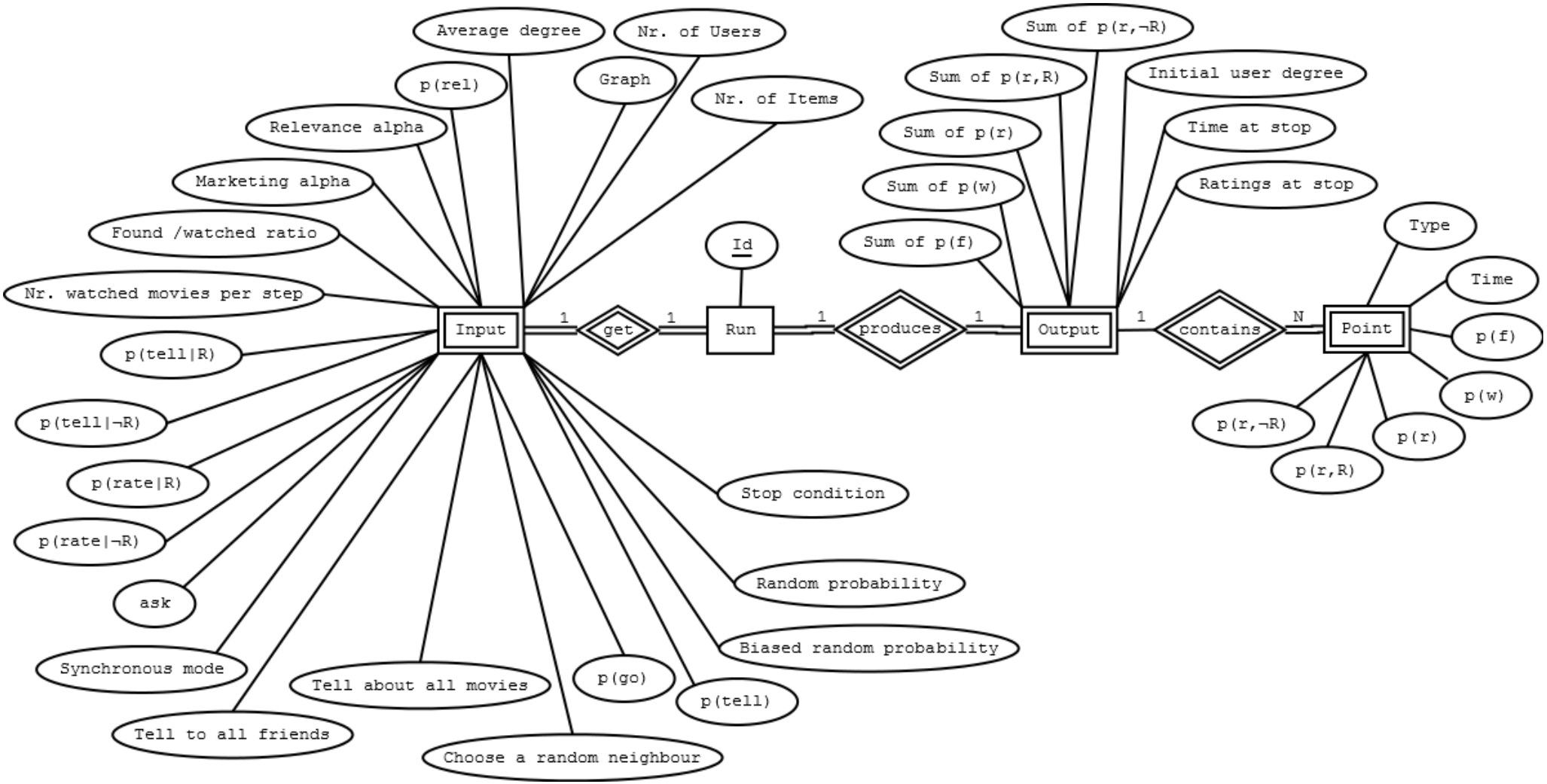


Figura 12. Diagrama ER de la base de datos.

5.3 Interfaz de usuario

El modelo y las funcionalidades de simulación descritas en los Capítulos 3 y 4 se han dotado de una interfaz gráfica orientada a la visualización detallada de la evolución de las simulaciones y los estados resultantes. Se han seleccionado para ello las variables de interés, de acuerdo con los objetivos que mencionan este trabajo, que se visualizan en diversas gráficas dinámicas que evolucionan en el tiempo, con diversas posibilidades para que el usuario avance, detenga o retroceda en la visualización en cualquier momento.

La interfaz se puede observar en la Figura 13. En los apartados que siguen se describe cada una de las ventanas de las que se compone dicha interfaz.

5.3.1 Ventana principal (1-4)

En esta ventana se muestran los elementos principales del estado del sistema según avanza la simulación. Engloba varios paneles y elementos que se explican a continuación.

Gráfico de distribuciones (1)

Muestra en un eje de coordenadas las distintas curvas de salida explicadas en el apartado 4.2.5. El eje de abscisas representa los ítems y el eje de ordenadas el ratio de usuarios que la han descubierto/consumido/votado. Los ítems del eje *x* se pueden ordenar de mayor a menor por cualquiera de las distribuciones que se consideran en esta gráfica, seleccionando el radio botón correspondiente que se encuentra a la izquierda del nombre de cada distribución, en la leyenda de la zona superior. De modo que cuando se ordena por una de las distribuciones, su curva se hace monótona decreciente (con forma aparente de curva más definida), y las demás se ven como nube de puntos, más o menos borrosa según la distribución correlacione menos o más, respectivamente, con la distribución que determina el orden. Además, se pueden seleccionar y deseleccionar las curvas que se desean ver, para aislar y comparar determinadas curvas, o visualizar varias o todas al mismo tiempo.

Observamos que la leyenda con los nombres de las distintas distribuciones se encuentra dividida en dos. Las curvas de la izquierda son las generadas por la simulación y las de la derecha (de color morado) proceden del conjunto de datos reales que se esté analizando en ese momento. Si no se está procesado ningún dataset, estas últimas distribuciones no aparecen. Tampoco lo hace el botón Final Plots, que se encuentra junto a la leyenda y que indica si se muestra la distribución final o la evolución temporal de las curvas procedentes de ratings reales.

Para facilitar la visualización de aquellas curvas con valores muy bajos que apenas despejan del eje *x* se ha incorporado la opción de hacer Zoom. Cuando se selecciona este botón, el máximo valor del eje *y* deja de ser uno y pasa a ser el valor más alto de las curvas que se están mostrando en ese momento. Esto reescala las curvas y las amplía hasta rellenar todo el plano.

Barra de progreso y controles de tiempo (2)

En la zona 2 se encuentra la barra de progreso, que indica la unidad de tiempo que se está visualizando, y unos controles de tiempo que permiten detener y reanudar la simulación, así como desplazarse en el tiempo. También la barra de progreso permite el desplazamiento en el tiempo con el movimiento manual del señalizador.

Promedios (3)

En la zona 3 se representa la evolución en el tiempo de los promedios de las distintas distribuciones de salida. Es decir el eje *x* representa el tiempo, y el eje *y* el promedio sobre todos los ítems de la curva del mismo color en el área 1. Por ejemplo, la curva verde del área 1 representa el ratio de usuarios que han descubierto cada ítem, y la curva verde en la zona 3 muestra el promedio por ítem de esa curva (o si se prefiere, el área bajo la curva, dividida por el nº de ítems).

Evaluación de los recomendadores (4)

En la zona 4 se muestra la evaluación de una serie de recomendadores mediante distintas métricas. Esto es objeto de trabajo futuro y por ello no se explica más en detalle. La elección sobre qué evaluación de qué recomendador mostrar se realiza en la ventana de selección de métricas y recomendadores (zona 5) qué, además, funciona a modo de leyenda indicando el color de la cada curva.

5.3.2 Ventana de parámetros (6)

En la ventana señalizada con el número 6 se muestran y ajustan todos los parámetros de la simulación. Se permite modificar dichos parámetros una vez iniciada la simulación, en cuyo caso las siguientes iteraciones se realizarán con los nuevos valores.

5.3.3 Visualización de la distribución del grado (8)

Lo más común es representar las distintas distribuciones indicando para cada ítem el ratio de usuarios que han interactuado de cierta forma con él. Sin embargo, en el caso de las tres distribuciones de ratings, vamos a considerar además, lo que se denomina distribución del grado, esto es, para cada número de ratings, el número de ítems que han recibido tantos ratings.

Esta distribución es la que se muestra en la ventana señalada con el número 8 con el eje de ordenadas en escala logarítmica. Proporciona una vista complementaria a la distribución ítem a ítem (zona 1), y permite en particular apreciar mejor las distribuciones power law, que presentan forma de tendencia lineal en esta escala.

5.3.4 Ventana de ajuste (7)

Cuando se opta por introducir unos ratings reales con los que comparar los resultados del modelo, se muestra la ventana señalada con el número 7. Si no se introducen dichos ratings, esta ventana no aparece al igual que tampoco lo hacen las curvas del gráfico de distribuciones que hacen referencia a los datos de votos reales.

Esta ventana muestra la evolución en el tiempo de la diferencia entre las gráficas reales y las simuladas, según la medida que se haya seleccionado (KLD, error medio…), de acuerdo con lo descrito en la sección 4.4.1. En el caso de la Figura 13 se muestra la evolución del error medio.

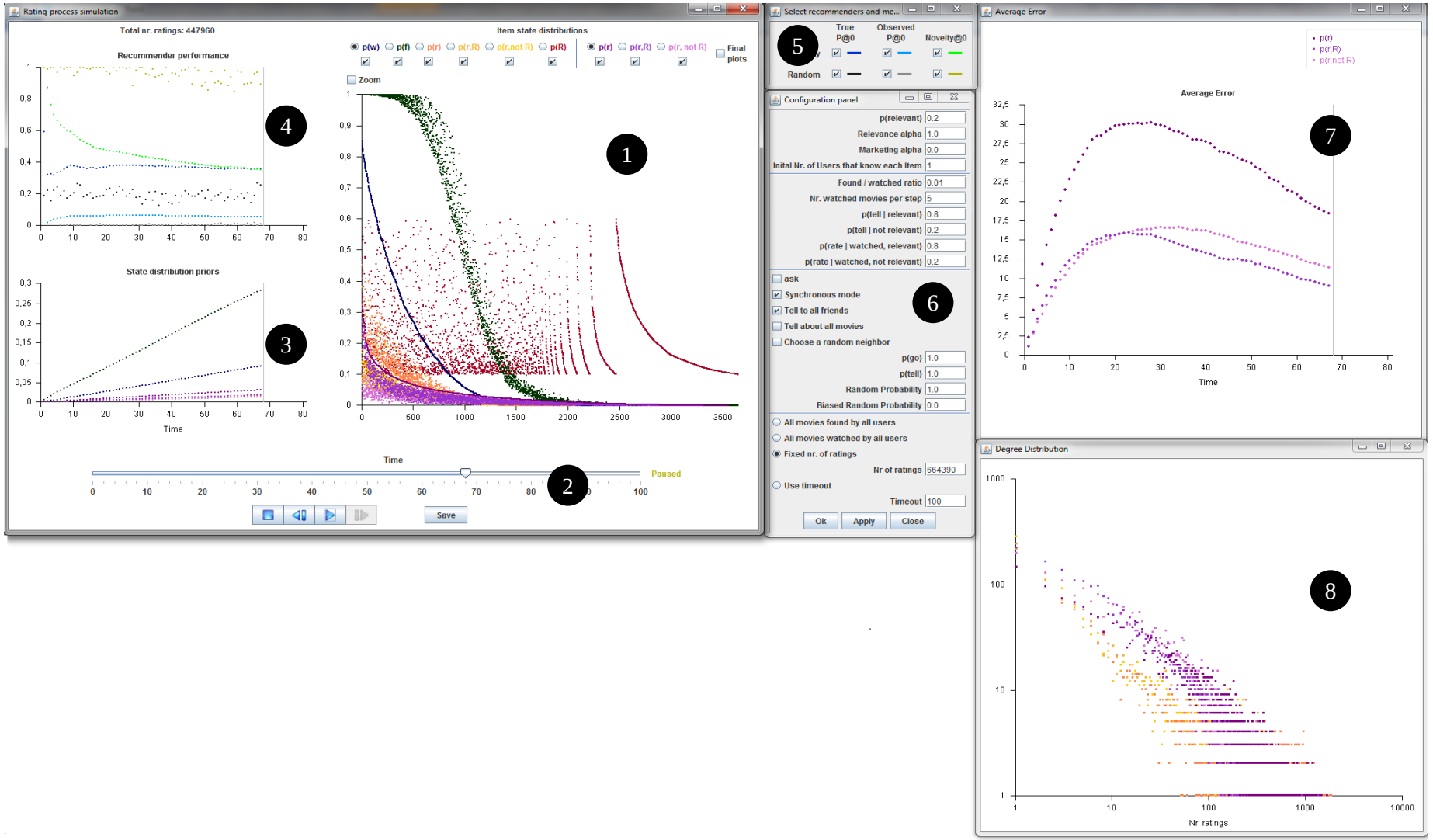


Figura 13. Intefaz de la aplicación principal.

5.4 Visualización de la propagación en grafo

Con el objetivo de observar de forma cualitativa y en detalle el proceso de propagación a través del grafo de relaciones sociales, se ha realizado un programa auxiliar que sigue el modelo anterior pero en el que sólo hay un ítem. Dicho programa muestra el grafo de la red social e indica de forma gráfica el estado de cada usuario, esto es: si ha descubierto el ítem, si lo ha consumido, si lo ha votado… También señala el nodo que tiene el turno y cómo interacciona con los demás en el proceso de comunicación.

Una primera diferencia con la interfaz de usuario descrita en la sección anterior es pues que en la que ahora vamos a describir existe un solo ítem, mientras que en la anterior se puede fijar cualquier número de ítems (también uno si se desea). Pero la diferencia más importante es que la interfaz anterior muestra información agregada, mientras que ésta la muestra más detallada. Concretamente, en la primera interfaz se visualizan los promedios de cada estado (descubierto, puntuado, etc.) de cada ítem sobre todos los usuarios, mientras que en la que ahora vamos a describir se muestra, para un único ítem, el estado de cada usuario (lo ha descubierto, le gusta, lo ha puntuado, habla de él con un vecino, etc.) paso a paso.

Por otro lado, la interfaz anterior está desarrollada en Java, mientras que ésta lo está en JavaScript ejecutable desde una página web. El programa está disponible para su ejecución desde la dirección http://ir.ii.uam.es/~rocio/ItemPropagation/.

En la Figura 14 observamos la interfaz que se ha realizado con Javascript, JQuery, CSS y la librería D3[8] para representar grafos. Vemos a continuación cada una de las zonas numeradas de la figura.

Grafo, barra de progreso y controles de la simulación (1)

En la zona 1 observamos cuatro elementos: el grafo de la red social, la barra de progreso, los controles de la simulación y un conjunto de tres botones. Cada elemento se explica a continuación:

- Grafo: La mayor parte de la zona 1 se dedica a la visualización del grafo que representa la red social, formado por un conjunto de nodos y aristas. El tamaño de cada nodo depende del número de usuarios con los que está conectado. Cuantos más vecinos, mayor es dicho tamaño. El color del interior del nodo indica el estado en el que se encuentra el usuario, a saber:
  - Gris: no ha descubierto todavía el ítem.
  - Verde: ha descubierto el ítem pero todavía no lo ha consumido.
  - Morado: ha consumido el ítem pero todavía no lo ha votado.
  - Amarillo: ha votado el ítem.

  Cada uno de los anteriores colores puede tener una tonalidad oscura o clara que indica, respectivamente, si el ítem es relevante o no para ese usuario.

  El nodo que tiene el turno en cada momento se distingue por ser el que tiene el borde rojo. En la Figura 14, dicho nodo se observa arriba a la izquierda con un color amarillo, es decir, ha votado el ítem. Los arcos pintados de rojo destacan las conexiones del usuario que tiene el turno con sus vecinos.

[8] https://github.com/mbostock/d3/wiki/Gallery

La utilización de CSS facilita el cambio de estado y su visualización, pues basta con asignar al nodo una clase distinta.

- Barra de progreso: Inmediatamente por debajo del grafo se encuentra la barra de progreso de la simulación que indica la iteración (o lo que es lo mismo, la unidad de tiempo) en la que ésta se encuentra. En esta barra podemos observar dos señalizadores: el de color gris, que se encuentra más adelantado (entre la iteración 1400 y la 1600), informa del número de iteraciones que se han ejecutado hasta el momento y el de color azul (entre la iteración 200 y la 400) indica la iteración que se está visualizando. Es decir, el primero indica lo que está ocurriendo y el segundo lo que se está viendo. Ambos avances están desacoplados porque existen dos procesos independientes, uno que se encarga de ejecutar la simulación y guardar los cambios en los estados de los usuarios y otro que va leyendo esos cambios y visualizándolos. Lógicamente, el proceso de visualización es más lento y debe ir siempre por detrás. Así, como en un reproductor de streaming, podemos desplazar el punto de reproducción (segundo señalizador), entre la iteración cero y el punto de ejecución para visualizar el momento que creamos conveniente. Se puede controlar la velocidad a la que avanzan ambos señalizadores con las barras de la zona 4, que se explican más abajo en dicho punto.

  Cabe destacar que en este programa la unidad de tiempo se modifica ligeramente, pasando a representar un turno de un usuario en vez de la intervención de $q \cdot \mathcal{U}$ usuarios, como se establecía en la interfaz descrita en la sección anterior.

- Controles: debajo de la zona de progreso, situados ligeramente a la izquierda, se encuentran los controles de la simulación: stop, backward, play y forward, con la funcionalidad que su respectivo nombre indica.

- Botones: a la derecha de los controles de vídeo, se encuentran tres botones que permiten alterar la visualización del grafo que representa la red social de distintas formas:

  - Adjust: la zona en la que se visualiza el grafo permite acercar o alejar el punto de mira con el ratón y desplazar el grafo por la ventana. El botón Adjust sirve para volver a colocarlo en su posición inicial, esto es, centra el grafo en la ventana con un tamaño adecuado de visualización.

  - Stop/Start Layout: la posición de los nodos en la ventana puede venir fijada en el fichero del que se carga el grafo o puede ser necesario ejecutar un algoritmo de layout que determine dichas posiciones. Los layouts suelen ser bucles en los que las posiciones de los nodos se van reajustando unas en función de otras hasta que se alcanza una cierta estabilidad, pero dicha estabilidad podría no ocurrir nunca o tardar mucho en llegar. Por ello, este botón permite parar o reanudar el algoritmo de layout de forma que sea el usuario quien determine el punto en el que se alcanza un nivel óptimo de visualización.

  - Select new graph: permite seleccionar otra red social sobre la que ejecutar la simulación.

Promedios (2)

Las curvas que se muestran en esta zona son exactamente las mismas que las que se muestran en la zona 3 del programa descrito en la sección anterior, es decir, se muestran los promedios por ítem de los distintos ratios de estados de los usuarios respecto al ítem.

Sólo que en este caso, al tratarse de un solo ítem, el promedio es el mismo valor que presenta el único ítem en cuestión.

Parámetros (3)

Tabla en la que se ajustan los parámetros de la simulación, de igual forma que en la zona 6 del programa anterior. Si se modifica uno de estos parámetros durante la ejecución, las nuevas iteraciones se realizarán con la nueva configuración de parámetros. Para visualizar estas nuevas iteraciones deberemos desplazar el punto de visualización más allá de donde se encontraba el punto de ejecución cuando realizamos el cambio de parámetros.

Velocidad de la barra de progreso (4)

Esta zona muestra dos barras de progreso, *Play Delay* y *Run Delay*, que representan respectivamente los retardos introducidos a la hora de visualizar y ejecutar la simulación. Cuanto mayor sea el valor de la barra (o el del campo de texto correspondiente), mayor será el retardo y el correspondiente señalizador de la barra de progreso de la simulación (zona 1, debajo del grafo) se desplazará más despacio. Con estos ajustes el usuario puede adaptar la visualización a la velocidad que mejor se adecúe su capacidad de seguimiento visual y el nivel de detalle que se desee para apreciar los eventos de la simulación más o menos despacio.

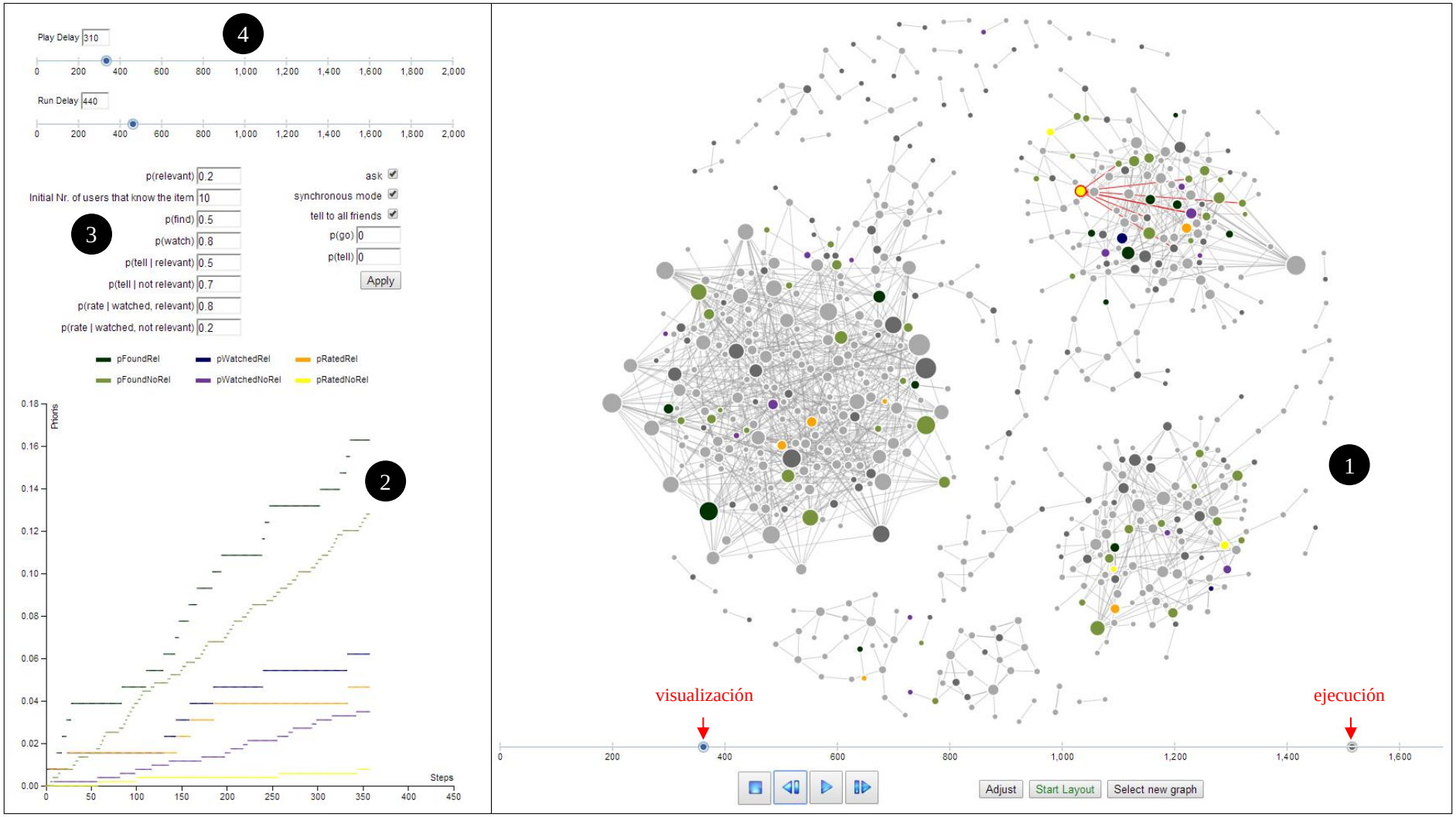


Figura 14. Intefaz de la visualización cualitativa de la propagación en grafo.

5.5 Librerias

El programa principal ha sido implementado usando el lenguaje de programación Java y haciendo uso de las siguientes librerías:

- Swing[9]: biblioteca gráfica para Java con la que se han implementado las interfaces de usuario, incluidas las gráficas de visualización punto por punto.
- Jung: también es una librería para Java y ofrece una gran variedad de métodos para modelizar y generar grafos. Se ha utilizado en la generación de los grafos Barabási y Erdös, así como soporte para la carga y recorrido de todos los grafos (también los externos), a excepción de los grafos (Twitter y Orkut, como veremos en la próxima sección) que por su gran escala han precisado una implementación propia optimizada al efecto.
- JRI: librería de Java que permite ejecutar código R. Se ha utilizado para aplicar el test de bondad de ajuste de Kolmogorov–Smirnov.
- SNAP: escrita en C++ y creada por la universidad de Stanford. Esta librería genera grafos aleatorios de tipo Barabási y Erdös con un rendimiento mucho mayor que Jung, por lo que se ha utilizado cuando se necesitaban grafos del orden de millones de usuarios. SNAP no se ha integrado con el programa, si no que los grafos generados por dicha herramienta se han volcado en fichero para, a continuación, leer dicho fichero desde nuestra aplicación.

Para visualizaciones rápidas de grandes grafos se ha utilizado el programa Gephi[10] pero sin integrar con la aplicación, únicamente a modo de herramienta auxiliar.

La visualización cualitativa de la red social se ha implementado en Javascritp, integrado con JQuery y CSS. Para la visualización y gestión del grafo y la ejecución del layout se utiliza la librería D3. Antes de elegir Javascript, se evaluaron otras opciones que finalmente se descartaron por su bajo rendimiento para visualizar grafos más o menos grandes:

- iGraph[11]: librería de Python para la visualización y gestión de grafos.
- Jung como visualizador de grafos.
- Gephi: en concreto una API que ofrece para Java[12].

Respecto a la base de datos, ésta se ha implementado en MySQL y se ha integrado con el programa Excel.

[9] http://docs.oracle.com/javase/7/docs/technotes/guides/swing/
[10] https://gephi.org/
[11] http://igraph.org/
[12] https://gephi.org/toolkit/

6. Experimentos

En este apartado se explican los experimentos realizados sobre la implementación del modelo explicado en los capítulos anteriores. En primer lugar, y a modo de validación del modelo propuesto, se contrastan los resultados obtenidos en la simulación con los publicados en el estudio de Doerr (2012) y con el modelo teórico de propagación de epidemias, ambos explicados en la sección 2.2 del estado del arte. Como vamos a ver, se comprueba empíricamente que, bajo cierta configuración de parámetros, ambos modelos son casos particulares del aquí propuesto. En segundo lugar se muestra el ajuste de algunos conjuntos de datos públicos y se indican las configuraciones de parámetros que hacen que obtengamos las distribuciones observadas en dichos datasets. Por último, se exploran y analizan algunos parámetros para determinar la forma en la que influyen en las derivas del sistema social y las distribuciones de ratings resultantes.

Respecto a las características de los equipos en los que se han llevado a cabo los experimentos, la mayoría se han realizado en un PC con un procesador de 4 GHz y 32 GB de memoria RAM. También se ha dispuesto de un servidor con 64 procesadores de 1.4 GHz cada uno y 512GB de memoria RAM, aunque al ser compartido hemos limitado el uso de la memoria a 50GB. Sin embargo, pese a la gran capacidad en memoria RAM de los equipos, ha sido este recurso el cuello de botella de la aplicación, sobre todo al procesar los conjuntos de datos públicos que, como veremos, ha sido necesario filtrar.

6.1 Reproducción del experimento de Doerr (2012)

Como validación de los resultados obtenidos en la simulación, se han contrastado éstos con los presentados en el reciente estudio por Doerr et al (2012), que se describió brevemente en la sección 2.2.1. El modelo desarrollado en el presente trabajo generaliza al de Doerr et al, por lo que particularizando nuestro modelo con la configuración que lo reduce al modelo de Doerr debería replicar un resultado equivalente al descrito por estos autores.

Antes de mostrar dicha configuración de parámetros, veamos detalladamente las innovaciones que presenta nuestro modelo frente al propuesto por Doerr et al.

- Presencia de la relevancia en el proceso de comunicación. Mientras que en el modelo de Doerr la transferencia de información se produce instantáneamente una vez que dos individuos contactan, en nuestro escenario esta comunicación depende de lo relevante que resulte dicha información para el usuario que está hablando, el cual puede decidir no transmitirla.
- Modos de comunicación más variados. Doerr et al consideran únicamente conversaciones entre dos individuos (se contacta con un solo amigo) en los que ambos interlocutores intervienen (contar y preguntar), mientras que en nuestro caso se tiene todo un abanico de combinaciones respecto al modo de comunicación tal y como se describe en la sección 4.2.4.
- Escenario de generación de ratings, no sólo de propagación de información. Doerr et al están únicamente interesados en los fenómenos de propagación en sí, mientras

que nosotros tenemos también interés por ver la relación de estos fenómenos con la generación de ratings, motivo por que se incorporan al escenario todos los elementos necesarios para realizar votaciones.

Veamos ahora la configuración de parámetros que da lugar a la equivalencia entre ambos modelos:

- Sólo se considera un ítem (*Nr. of Items* = 1).
- No hay descubrimiento exógeno y toda la propagación de información recae sobre la red social (*Found /watched ratio* = 0).
- Inicialmente el ítem lo conoce un usuario elegido al azar (*Initial Nr. Users* = 1).
- En cada iteración se recorren todos los usuarios secuencialmente y cada usuario elige un vecino al azar con el que comunicarse (*Choose a random neighbor* = verdadero). Los usuarios se comunican una vez por cada turno y no únicamente después de consumir el ítem (*Synchronous mode* = falso)
- La comunicación fluye en ambos sentidos, esto es, cuando un usuario habla con otro, además de hablarle de la película si la ha visto, también le pregunta por ella (*ask* = verdadero).
- No se considera la variable relevancia. Cuando hay una conversación, si alguno de los dos usuarios conoce el ítem, el otro lo descubre automáticamente, sin que ninguna probabilidad ni relevancia medie en el proceso.

  Una forma, entre otras, de anular la influencia de la relevancia en la simulación es hacer que el ítem sea relevante para todos los usuarios ($p(rel)$ = 1 y *Relevance alpha* = 0). Para anular la probabilidad de comunicar, hacemos que los usuarios hablen de la película siempre que sea relevante ($p(tell|R) = 1$). Como la película siempre es relevante, cuando les toque el turno, hablaran de ella con probabilidad 1.
- La simulación finaliza cuando todos los usuarios conocen el ítem (*All movies found by all users* = verdadero).

En el estudio de Doerr, se compara la velocidad a la que los usuarios descubren el ítem cuando se considera un grafo procedente de una red social real, en concreto se estudian las de Twitter y Orkut cuyos datos pueden observarse en la Tabla 6, con grafos generados artificialmente: Barabási, Erdös y el grafo completo. Así pues, hemos ejecutado nuestro modelo con configuración equivalente, sobre estos mismos grafos.

Red social	Nº. de usuarios	Nº. de conexiones
Orkut	3.072.441	117.185.083
Twitter	52.579.682	1.963.263.821

Tabla 6. Datos volumétricos de las redes sociales Orkut y Twitter.

Como observamos en la Tabla 6, las redes sociales de Orkut y Twitter son muy grandes – sobre todo Twitter – y los grafos de Jung en los que se soportan dichas redes necesitan más memoria de la disponible. Así, para poder tratar con grafos tan grandes, hemos diseñado la estructura ArrayGraph explicada en la sección 5.1.2. Con ella, se han conseguido reproducir todas las gráficas para la comparativa de Orkut. Sin embargo, en la comparativa con Twitter únicamente hemos podido reproducir las gráficas correspondientes al grafo de Twitter y al grafo completo debido a que no se ha

encontrado ninguna herramienta que consiga generar los grafos Barabási y Erdös correspondientes a tal cantidad de usuarios, en los equipos disponibles para nuestro trabajo. Para conseguir el mismo grafo de Twitter que utilizan Doerr et al en su experimento, contactamos con los autores de la página *The Twitter Project Page at MPI-SWS*[13] quienes nos lo facilitaron.

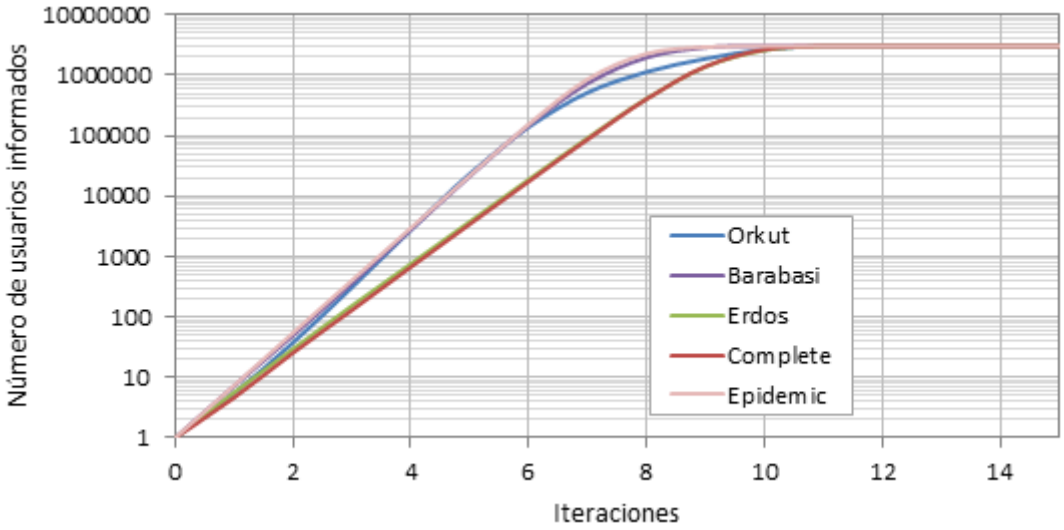


Figura 15. Comparativa con la red social Orkut obtenida al reproducir el experimento de Doerr et al.

La Figura 15 muestra el resultado sobre el grafo de Orkut, reflejando el número de usuarios que han descubierto el ítem por unidad de tiempo. La gráfica incluye también una curva teórica de propagación (Epidemic) que se explicará en el siguiente apartado. Observamos que el comportamiento es prácticamente idéntico al descrito por Doerr et al. Las gráficas correspondientes al grafo Erdös y al grafo completo muestran la misma evolución que en el artículo, prácticamente una encima de la otra, y lo mismo ocurre con las de Barabási y Orkut, quedando Orkut un poco por debajo en las últimas iteraciones.

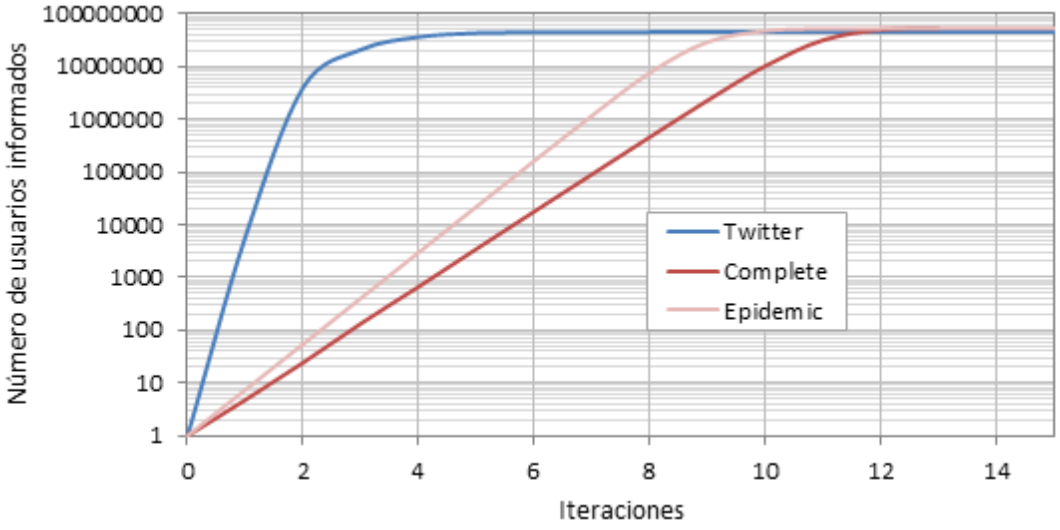


Figura 16. Comparativa con la red social Twitter obtenida al reproducir el experimento de Doerr et al.

[13] http://twitter.mpi-sws.org/

La comparativa entre el grafo completo y el grafo de Twitter se muestra en la Figura 16 en donde se observa que, nuevamente, las curvas presentan una forma muy similar a la descrita por Doerr et al.

En ambos experimentos, la única diferencia observada es que las gráficas de la simulación quedan ligeramente por encima de las del artículo, indicando que la simulación es algo más rápida. Tras analizar largamente en detalle y profundidad las posibles causas (algorítmicas, del modelo teórico, etc.) y variando todos los aspectos posibles como la elección del vecino o la forma de recorrer los usuarios, consideramos que la ligera diferencia puede deberse a detalles implementativos que no se exponen en dicho artículo.

6.2 Reproducción de una epidemia.

A modo de validación, contrastamos también los efectos de nuestro modelo con los modelos simples de propagación de epidemias, teniendo en cuenta la equivalencia que hace que estos últimos sean un caso particular de nuestro modelo. Como ya se explicó en el apartado 2.2.2, la ecuación que modeliza la propagación de una epidemia según el modelo IS que mide el número de personas infectadas en función del tiempo es la siguiente:

$$x(t) = \frac{x_0 e^{\beta t}}{N - x_0 + x_0 e^{\beta t}}$$

Donde x_0 es el número inicial de infectados, β el número promedio de contactos por persona y unidad de tiempo, y N el número total de personas en la población donde se propaga la epidemia.

Observamos que el modelo de Doerr et al, y por tanto el caso particular correspondiente de nuestro modelo, tiene equivalencia con un proceso de epidemia de la siguiente forma:

- El ítem es la enfermedad.
- Los usuarios que lo conocen son los infectados y los que no lo conocen los susceptibles.
- La comunicación entre dos usuarios equivale a un contacto en el modelo de epidemia. Cuando uno de ellos conoce el ítem y el otro no (uno está infectado y el otro no), el que no lo conoce lo descubre (se contagia).
- La red social es un grafo completo porque la ecuación se deduce asumiendo que todos los individuos pueden encontrarse con cualquier otro.
- Como inicialmente el ítem sólo es conocido por un usuario, tenemos $x_0 = 1$.

Para conocer el número de contactos por usuario tenemos en cuenta lo siguiente: todos los usuarios contactan con otro al menos una vez, la correspondiente al turno del usuario, y el número de veces que un usuario es contactado por otro sigue una distribución binomial con parámetros $n = N - 1$ y $p = \frac{1}{N-1}$. Por tanto, en promedio cada usuario es contactado una vez ya que:

$$E(B(n,p)) = n \cdot p = (N-1) \cdot \frac{1}{N-1} = 1$$

En total se tiene que:

$$\beta = 1 + E\big(B(n,p)\big) = 1 + 1 = 2$$

Por tanto, la curva empírica del grafo completo debería, teóricamente, ajustarse a la ecuación:

$$x(t) = \frac{e^{2t}}{N - 1 + e^{2t}}$$

En las Figuras 15 y 16 podemos observar los valores de esta ecuación (bajo el nombre de *Epidemic*) para el número de usuarios de Orkut y Twitter respectivamente. En ambas gráficas dicha curva queda por encima de la obtenida en la simulación para el correspondiente grafo completo. El motivo de esta diferencia se debe al error de aproximación que se introduce en la derivación de la formula teórica. En concreto, al considerar el número de infectados se deduce una ecuación recursiva $x_n = f(x_{n-1})$ donde f es un polinomio de grado dos en la variable x_{n-1} cuyo coeficiente del término cuadrático es negativo. Al promediar para conocer el número medio de infectados por iteración se hace $E(x_n) = f\big(E(x_{n-1})\big)$, es decir, se asume la igualdad $E(x_{n-1}^2) \sim E(x_{n-1})^2$ cuando en general es $E(x_{n-1}^2) \geq E(x_{n-1})^2$ o, lo que es lo mismo, $E(-x_{n-1}^2) \leq -E(x_{n-1})^2$. Por tanto, en general se tiene $E(x_n) \leq f\big(E(x_{n-1})\big)$ y el incremento aproximado es ligeramente superior al real, tal y como se observa en las Figuras 15 y 16.

6.3 Ajuste de datos reales

Tal y como se explicó en la sección 1.2, uno de los objetivos del trabajo es generar distribuciones de ratings como las que se observan en datasets reales y así, además de contrastar la corrección del modelo, poder conocer la configuración de parámetros que da lugar a dichas distribuciones y deducir las dinámicas que han intervenido en la generación real de los ratings. Con este objetivo en mente, hemos buscado el ajuste mediante nuestro modelo de las distribuciones observadas en varios datasets. Concretamente, se han tomado datos de MovieLens, Epinions, Twitter, y Foursquare.

Los ratings de estos cuatro datasets poseen una marca temporal que permite ver la evolución en el tiempo de las distintas curvas de ratings – $p(rate)$, $p(rate, relevant)$ y $p(rate, not\ relevant)$ – y compararlas con las producidas por la simulación. Este dato, junto con el hecho de que tenemos varias curvas de ratings susceptibles de ser ajustadas, nos sugiere distintas opciones cuando hablamos de ajustar.

- **Ajuste de una curva vs. ajuste de varias curvas a la vez.** Como ya se ha explicado anteriormente, estamos considerando tres curvas referentes a la distribución de ratings: $p(rate)$, $p(rate, relevant)$ y $p(rate, not\ relevant)$. Nuevamente se nos plantea la opción de ajustar las tres curvas a la vez frente a sólo ajustar una o dos de ellas.

  Cabe destacar que si ajustáramos las curvas $p(rate, relevant)$ y $p(rate, not\ relevant)$ automáticamente también quedaría ajustada la curva $p(rate)$, pues es la suma de las dos anteriores. La implicación no funciona en el otro sentido, esto es, ajustar $p(rate)$ no implica que las otras dos curvas también se ajusten, pues se podrían compensar las diferencias. Sin embargo, en los casos estudiados se ha visto que al ajustar $p(rate)$ las curvas $p(rate, relevant)$ y $p(rate, not\ relevant)$ no presentan diferencias significativas entre las simuladas y

las extraídas del dataset. Por ello, hablaremos únicamente del ajuste de la curva $p(rate)$.

- **Ajuste final vs. ajuste temporal.** Normalmente, cuando hablamos de encontrar la configuración de parámetros que mejor ajusta la distribución de ratings de un dataset nos referimos a la distribución final, sin tener en cuenta los momentos en los que se realizan los votos. Sin embargo, también es posible considerar para cada instante la distribución de los ratings realizados hasta ese momento. Así, tendríamos una curva para cada tiempo t y estaríamos interesados en ajustar todas estas curvas a la vez.

  El ajuste temporal es, por tanto, mucho más exacto que el final, porque no solamente simula la última distribución sino también la forma en la que se ha llegado hasta ella. Sin embargo, ajustar varias curvas a la vez es considerablemente más complicado que ajustar una sola, ya que parámetros que ajustan bien los momentos finales de la simulación podrían no predecir correctamente la evolución de los primeros instantes y viceversa. Una posible solución sería promediar las diferencias entre las distintas curvas y encontrar la configuración de parámetros que minimiza este promedio. Sin embargo, se ha decidido por simplicidad ajustar únicamente la distribución de ratings final de la curva $p(rate)$, dejando un ajuste mucho más evolucionado para posibles trabajos futuros.

Respecto a la forma de llevar a cabo los ajustes, en general se ha realizado de manera informal, partiendo de unos valores base y variando a mano los parámetros hasta encontrar una configuración más o menos satisfactoria. En algún caso, tras alcanzar dicha configuración, se han realizado barridos sistemáticos de los parámetros en torno a ella para elegir a continuación los valores que menor error medio presentaban con la curva de ratings del dataset. Un estudio más formal y sistemático es trabajo futuro una vez consolidada la plataforma desarrollada.

En los siguientes apartados, se explican en primer lugar las restricciones impuestas en el tamaño de los datasets debido a las capacidades limitadas de los equipos. A continuación se describen los cuatro datasets estudiados y los filtros necesarios para adecuarlos a dichas restricciones. Por último, se exponen los resultados de los ajustes.

6.3.1 Restricción en el tamaño de los datasets

La aplicación debe simular varias interacciones usuario-ítem, es decir, debe almacenar los ítems que son descubiertos, vistos y votados por cada usuario, así como los que le resultan relevantes. Esto implica que, con N usuarios y M ítems, el programa debe soportar varias matrices que potencialmente pueden llegar a tener un tamaño de $N \times M$ aunque en general será inferior, pues no todos los usuarios interaccionan con todas las películas.

Así, si $N \times M$ presenta un orden de magnitud superior a 10^7, (p.e. 10^8) tendríamos un coste en memoria para las matrices de interacción usuario-item del orden de $8(bytes\ por\ entero) \times 10^8(tamaño\ de\ la\ matriz) \times 4\ (número\ de\ matrices) = 1.6\ GB$. Si consideramos el resto de variables de la aplicación, incluidas las listas con las gráficas de la interfaz, podemos llegar a saturar la capacidad en memoria. Si añadimos un orden más ($N \times M \sim 10^9$) alcanzaríamos los 16 GB de memoria RAM.

Se ha comprobado experimentalmente que la aplicación presenta un buen rendimiento con datos en los que $N \times M$ es del orden de 10^7 o menor, pero superando estos órdenes la maquina se queda sin suficiente memoria.

Algunos de los datasets estudiados presentan tamaños con varios órdenes de magnitud por encima de 10^7 por lo que ha sido necesario filtrarlos para alcanzar valores más manejables.

En los siguientes apartados se explican las características de cada uno de los datasets, tanto antes como después de filtrarlos, y posteriormente los resultados de los distintos ajustes.

6.3.2 MovieLens

Tal y como se explicó en la sección 2.3.1, MovieLens es un conjunto de datos públicos que contiene puntuaciones de usuarios a películas. El dataset consta 6.000 películas, 4.000 usuarios y 1 millón de ratings. Como vemos, es un dataset de pequeño tamaño ($N \times M \sim 10^6$) y no es necesario filtrarlo.

A diferencia del resto de datasets, MovieLens no aporta información acerca de la red social que subyace tras la generación de ratings, esto es, de las relaciones entre los distintos usuarios. Por ello se ha optado por ejecutar la simulación sobre un grafo generado automáticamente según el modelo Barabási, dado que es el que más similitud presenta (salvando las distancias) con una red social real.

6.3.3 Epinions

Epinions[14] es un portal online en el que los usuarios reportan valoraciones subjetivas acerca de productos (o artículos) que han consumido. Además, unos usuarios pueden confiar (*trust*) o no (*distrust*) en otros usuarios de forma que la aplicación nos mostrará las nuevas crónicas de los usuarios en los que confiemos y no nos mostrará las de aquellos en los que no confiemos.

El dataset que se ha utilizado contiene 132.000 usuarios (aproximadamente), 841.372 conexiones (717.667 trusts y 123.705 distrusts), 1.560.144 artículos y 13.668.319 ratings. Vemos que es demasiado grande para ser soportado por la aplicación ($N \times M \sim 10^{11} \gg 10^7$) por lo cual hemos filtrado un conjunto más pequeño de la siguiente forma: en primer lugar se eliminan los ítems necesarios para que el número total de ítems se encuentre en torno a 7.000, eligiendo con mayor probabilidad los ítems con menos votos, a continuación se eliminan los usuarios que no realizan ningún voto sobre los ítems restantes que sobreviven al paso 1. Tras este filtrado se ha obtenido un dataset con 18.279 usuarios, 7.139 ítems y 425.060 ratings.

Respecto a los votos, en el dataset van de 1 a 5 por lo que, para ajustarlos al modelo binario, hemos considerado que los votos de 1, 2, 3 indican ítems no relevantes para un usuario, y los votos 4 y 5 indican relevancia. Por último, como red social hemos considerado los enlaces *trust*, pues indican que la información que publica un usuario llega a los usuarios que confían en él.

6.3.4 Twitter

La red social Twitter también puede interpretarse como una red social en la que existen ítems sobre los que se realizan votos. Ello se realiza "plegando" el espacio de ítems sobre el espacio de usuarios, dando a éstos un papel dual, como sigue.

- Red social: la estructura de seguidores y seguidos de Twitter constituye una red social dirigida en la que la arista (u, v) indica que el usuario u es seguido por el usuario v, y la información, es decir los tweets, fluyen del usuario u al usuario v.

[14] http://www.epinions.com/

- Ítems: los ítems son a su vez los propios usuarios, que de este modo juegan un papel dual como tales usuarios pero también ítems.
- Ratings: un usuario (ítem) recibe un voto positivo cuando uno de sus tweets es retweeteado. Los valores de ratings son binarios, y no se permiten votos negativos, lo cual implica que todos los ítems son relevantes para todos los usuarios, y los valores de rating sólo pueden ser 1 o desconocido.

Alternativamente, se podría naturalmente considerar que los ítems son los tweets, en vez de los usuarios que los escriben. Sin embargo, hemos descartado esta opción dado que pocos tweets obtienen un número de retweets significativamente grande frente al resto como para dar pie a un análisis interesante dada la falta de masa crítica de ratings que recibe cada ítem.

El dataset analizado se ha tomado del Trabajo de Fin de Grado de Alfonso Alhambra (Alhambra 2014). Consta de 10.029 usuarios, 2.028.097 tweets y 90.148 retweets. Al establecer que los usuarios son también los ítems, se tiene que $N \times M = N \times N \sim 10^8 > 10^7$. Aunque sólo ligeramente, el tamaño de este dataset también es superior al viable para nuestros recursos.

Para filtrarlo, hemos considerado únicamente los usuarios que participan en algún retweet, obteniendo así 8.877 usuarios, 5.960 ítems (usuarios cuyos tweets son retweeteados al menos una vez) y 90.147 ratings (retweets).

6.3.5 Foursquare

Foursquare[15] es una red social en la que los usuarios pueden hablar y puntuar lugares en los que han estado (*check-ins*: lugares visitados) y compartir dichas valoraciones con sus amigos. En función de sus intereses y los de sus amigos, la aplicación sugiere nuevos lugares a visitar que se encuentran cerca de la posición del usuario.

El dataset que se ha analizado[16] contiene 2.153.471 usuarios, 1.143.092 lugares (ítems), 1.021.970 *check-ins*, 27.098.490 conexiones entre usuarios y 2.809.581 ratings. En concreto, nos interesan dos elementos de esta colección: la red social y los ratings.

Respecto al tamaño del conjunto de datos, el número de usuarios e ítems es demasiado elevado ($N \times M \sim 10^{12} \gg 10^7$) por lo que hemos llevado a cabo el mismo filtrado que el realizado para procesar Epinions. Esto es, eliminamos los ítems con menos votos hasta dejar del orden de 7.000 ítems y a continuación eliminamos los usuarios que no realizan votos sobre el conjunto de ítems seleccionados. Como resultado, obtenemos un dataset con 168.810 usuarios, 6.435 ítems y 382.084 ratings.

Por otro lado, los ratings no son binarios, sino que toman valores de 1 a 5. La documentación adjunta a los datos no deja claro de dónde se obtienen estos ratings, porque en la aplicación se observa que los usuarios sólo pueden indicar si les gusta o no les gusta un cierto lugar (voto binario). Por otra parte, se observa una distribución muy poco frecuente: los ítems con más votos, tienen más votos negativos (3 o menos) que positivos (4 o 5). Dada la falta de documentación (tanto en la web de Foursquare como en foros) que aclare estas dudas, omitimos la distinción y consideramos como simplificación que todos los votos son relevantes. En la Tabla 7 se resumen los datos de los datasets tras los filtrados pertinentes.

[15] https://foursquare.com/
[16] https://archive.org/details/201309_foursquare_dataset_umn

Dataset	Nº. de usuarios	Nº. de ítems	Nº. de ratings
MovieLens	4.000	6.000	1.000.000
Epinions	18.279	7.139	425.060
Twitter	8.877	5.960	90.147
Foursquare	168.810	6.435	382.084

Tabla 7. Datos de los conjuntos de datos MovieLens, Epinions, Twitter y Foursquare tras el preprocesamiento realizado para ajustar los tamaños.

6.3.6 Resultados

El ajuste de la curva $p(rate)$ – ratio de usuarios que han votado cada ítem – de los conjuntos de datos explicados anteriormente se muestra en la Figura 17 y las configuraciones de parámetros que han dado lugar a dichos ajustes se encuentran en las Tablas 8 y 9. La primera contiene los parámetros cuyo valor es común al ajuste de todos los datasets, mientras que en la segunda se muestran los valores de los parámetros que difieren de un dataset a otro.

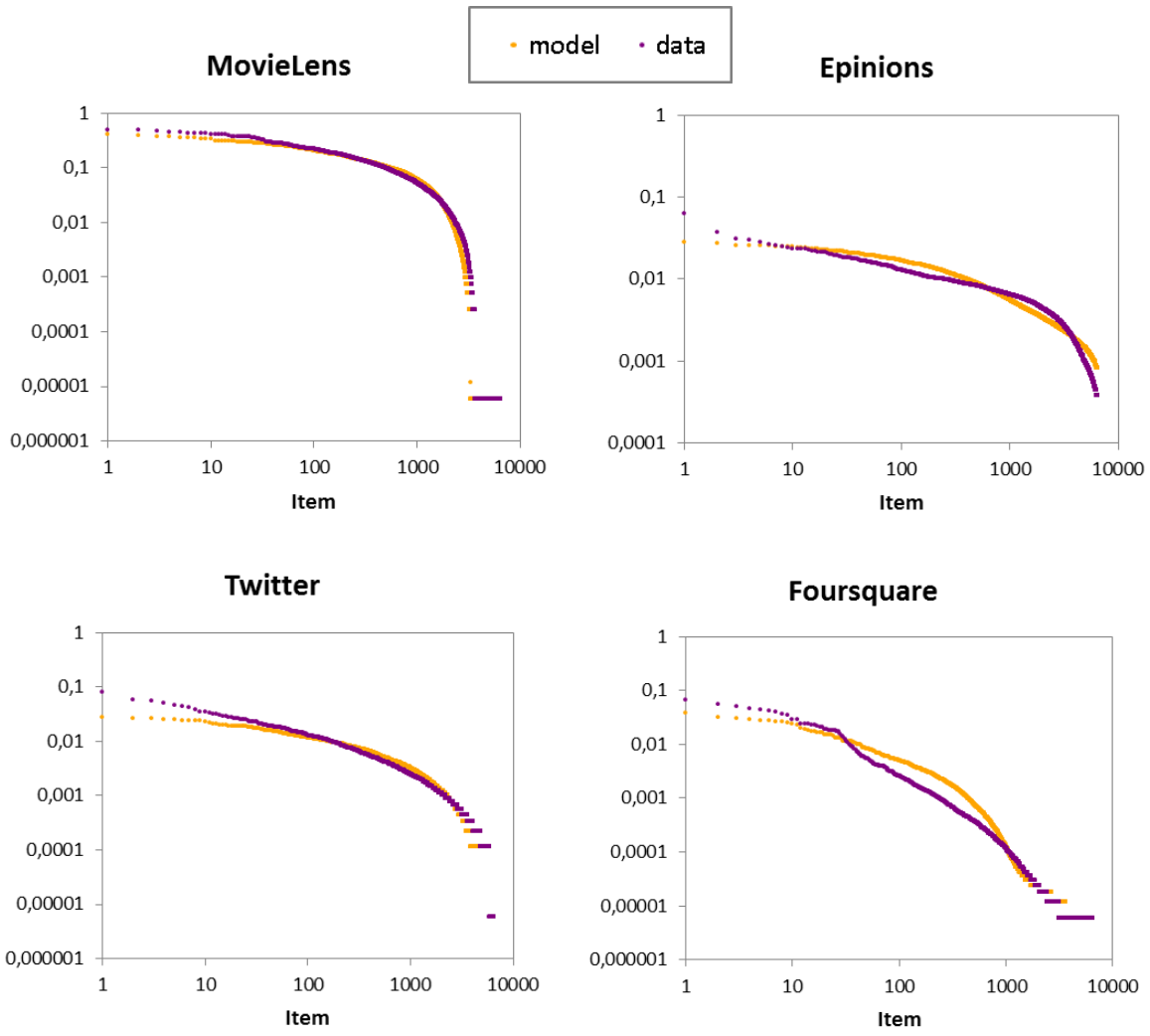


Figura 17. Ajuste de los conjuntos de datos públicos MovieLens, Epinions, Twitter y Foursquare.

Como ya se explicó al inicio de la sección, el ajuste se ha llevado a cabo de varias maneras. En el caso de MovieLens, donde tanteando con distintos valores se encontró en seguida una configuración que aproximaba la curva bastante bien, se realizaron barridos sistemáticos de los valores de los parámetros en torno a esa configuración encontrada y luego se eligió la configuración que menor error medio presentaba con la curva de ratings del dataset. Este barrido sistemático es la causa por la cual observamos que las curvas simulada y real se ajustan mejor en el caso de MovieLens que en el resto de datasets, con los cuales únicamente se tanteó hasta encontrar una configuración más o menos satisfactoria.

Respecto a los parámetros que ajustan cada dataset, cabe destacar que se han observado varias configuraciones con valores diferentes que ajustan de forma similar las curvas. Por ejemplo, tanto Twitter como Foursquare contienen únicamente votos positivos. En el caso de Twitter hemos simulado este hecho anulando todos los parámetros condicionados a la no relevancia, tanto referentes a la comunicación – $p(tell|\neg R)$ – como a la votación – $p(rate|\neg R)$ –, mientras que en Foursquare únicamente anulamos la creación de votos negativos – $p(rate|\neg R)$ – pero no la comunicación acerca de películas no relevantes.

Parámetro	Valor	Parámetro	Valor
$p(rel)$	0.2	$p(start)$	1
Relevance alpha	1	*Fraction of users per time step*	1
Nr. watched movies per step	5	*Random probability*	1
Choose a random neighbor	falso	*Biased Random Probability*	0
$p(go)$	1	*Marketing alpha*	0

Tabla 8. Asignación de parámetros cuyo valor es común al ajuste de todos los datasets.

En la Figura 17 vemos que la distribución de MovieLens presenta valores más elevados (máximo por encima de 0,1) que el resto de datasets (por debajo de 0,1). Por ello, los parámetros que influyen en la cantidad de votos que se generan – $p(tell|R)$, $p(tell|\neg R)$, $p(rate|R)$, $p(rate|\neg R)$ y el ratio entre películas vistas y descubiertas – tienen valores más altos en el ajuste de este dataset. Además, en todos los datasets los parámetros condicionados a la relevancia poseen un valor superior si el ítem es relevante que si no lo es, lo cual concuerda con (Steck 2010, 2011) que observa que en general, tendemos a manifestar con más probabilidad opiniones positivas que negativas.

En todos los casos excluimos la influencia de la publicidad, y dejamos que sea la red social y el descubrimiento exógeno los que condicionen la forma de la curva.

	Datasets			
Parámetros	**MovieLens**	**Epinions**	**Twitter**	**Foursquare**
Nr.of Items	Número de ítems del dataset			
Graph	Barabasi	Grafo del dataset		
Nr. of Users	Número de usuarios del dataset			

Average degree	40	Grado del grafo del dataset		
Found / watched ratio	0.02	0.001	0.001	0.015
$p(tell\|R)$	0.6	0.2	0.5	0.4
$p(tell\|\neg R)$	0.3	0.1	0	0.2
$p(rate\|R)$	0.6	0.2	1	0.4
$p(rate\|\neg R)$	0.2	0.01	0	0
ask	falso	verdadero	verdadero	falso
Synchronous mode	verdadero	verdadero	verdadero	verdadero
Tell to all fiends	verdadero	falso	falso	verdadero
Tell about all movies	falso	verdadero	verdadero	falso
Condición de parada	Número de ratings del dataset			

Tabla 9. Asignación de parámetros que difieren del ajuste de un dataset a otro.

6.4 Exploración de parámetros

Dada la gran cantidad de parámetros de la simulación, realizar un estudio exhaustivo de todas las posibles configuraciones y comportamientos queda fuera del alcance de este trabajo, y sería el tema de un posible trabajo futuro. Sin embargo, sí es interesante realizar unos primeros avances en este sentido, observando ciertos comportamientos generales y estudiando el efecto que produce la variación de ciertos parámetros.

En concreto vamos a analizar los siguientes aspectos:

- Relación entre comunicación y votación: vamos a estudiar cómo varía el ratio de votos realizados en función de los parámetros $p(tell|R)$y $p(tell|\neg R)$. Este experimento analiza la influencia de la elección de la película que vamos a comunicar en el número de votos totales que realizan los usuarios.
- Influencia del número de amigos con los que se comunica un usuario. Buscamos comparar sistemas de comunicación en publicación, tipo Twitter donde lo que un usuario dice llega a todos sus seguidores, con sistemas en que las conversaciones se producen únicamente entre dos individuos.
- Influencia del nivel de descubrimiento exógeno. Queremos comparar los efectos del descubrimiento por vía de la red social con los producidos al introducir sistemas exógenos de descubrimiento (publicidad, buscadores, recomendadores, etc.).
- Influencia del tipo de grafo. Nos interesa observar el efecto de la topología de la red social en la distribución del descubrimiento y demás interacciones usuario-ítem.

El motivo por el que hemos elegido estudiar las relaciones anteriores es porque suponen un subconjunto representativo de todos los parámetros de la simulación. A la hora de estudiar los aspectos anteriores vamos a variar ciertos parámetros dejando fijos otros. Para ello, tomamos como configuración de partida los valores de los parámetros en el ajuste de los datos de MovieLens.

6.4.1 Relación entre comunicación y votación

Los parámetros $p(tell|R)$, $p(tell|\neg R)$, $p(rate|R)$ y $p(rate|\neg R)$ son sencillos de estudiar y no requieren de experimento alguno. Si los incrementamos individualmente, incrementan de forma obvia, respectivamente, el descubrimiento relevante/no relevante y los ratings relevantes/no relevantes. Sin embargo, resulta interesante estudiar cómo al cambiar la relevancia de los ítems que se descubren por vía de la red social, determinada por los parámetros $p(tell|R)$ y $p(tell|\neg R)$, el número de votos puede variar pese a que el nivel de comunicación sea el mismo, esto es, manteniendo $p(tell)$ constante.

En primer lugar, veamos qué relación tienen que cumplir $p(tell|R)$ y $p(tell|\neg R)$ para mantener constante el nivel de comunicación que viene determinado por el priori $p(tell)$ (probabilidad de que un usuario hable de una película). Para ello, utilizamos la regla de la probabilidad total y tenemos que.

$$p(tell) = p(tell|R) \cdot p(R) + \ p(tell|\neg R) \cdot p(\neg R)$$

En el caso de MovieLens, con la configuración que se muestra en las Tablas 8 y 9 esta probabilidad tiene el siguiente valor:

$$p(tell) = 0.6 \cdot 0.2 + \ 0.3 \cdot 0.8 = 0.36$$

A continuación variamos los valores de $p(tell|R)$ y $p(tell|\neg R)$ de forma que $p(tell)$ se mantenga constante tal y como se indica en la Tabla 10:

$\boldsymbol{p(tell\|R)}$	$\boldsymbol{p(tell\|\neg R)}$
0	0,45
0,1	0,425
0,2	0,4
0,3	0,375
0,4	0,35
0,5	0,325
0,6	0,3
0,7	0,275
0,8	0,25
0,9	0,225
1	0,2

Tabla 10. Variación realizada de los parámetros $p(tell|R)$ y $p(tell|\neg R)$ de forma que el nivel de comunicación $p(tell)$ se mantenga constante.

Si ejecutamos la simulación para cada par de valores $p(tell|R)$ y $p(tell|\neg R)$ y observamos el valor de $p(rate)$ tras 200 iteraciones, obtenemos la curva que se muestra en la Figura 18, donde el eje *x* indica el valor de $p(tell|R)$. Observamos que según incrementamos el valor de $p(tell|R)$, aumenta también $p(rate)$ pese a que el descubrimiento total por vía de la red social $p(tell)$ no se incrementa. Es decir, se está descubriendo lo mismo, pero el número de películas votadas aumenta.

El motivo por el que ocurre esto es porque, aunque se descubre la misma cantidad de ítems, dicho descubrimiento no se distribuye de igual forma. Según aumenta el valor de $p(tell|R)$, se descubren más relevantes que no relevantes y la probabilidad de votar un relevante $p(rate|R) = 0.6$, es mayor que la de votar un no relevante $p(rate|\neg R) = 0.2$, por lo que el número de votos aumenta.

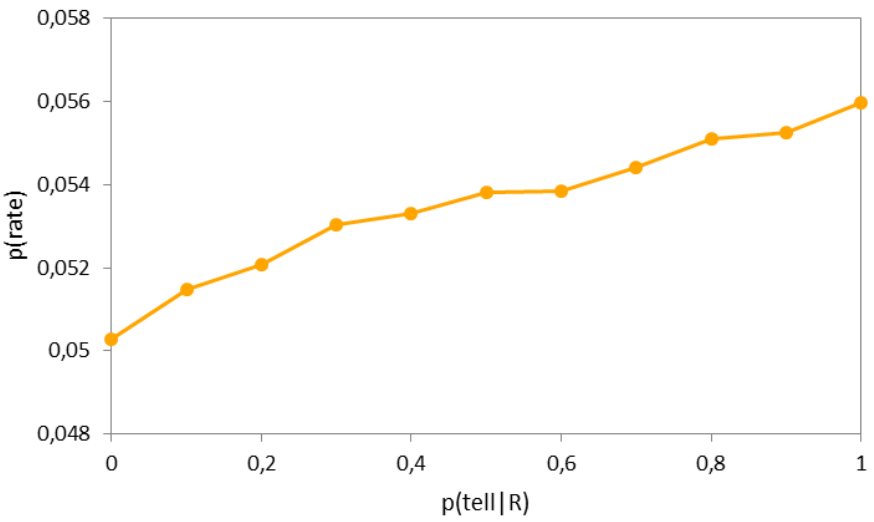


Figura 18. Valor de $p(rate)$ en función de $p(tell|R)$ pero manteniendo constante $p(tell)$.

6.4.2 Único amigo vs. todos los amigos

Si variamos el parámetro *Tell to all friends*, manteniendo el resto fijo, obtenemos dos curvas de descubrimiento muy diferentes, tal y como se observa en la Figura 19. Observamos que cuando los usuarios se comunican con todos sus amigos, el descubrimiento se vuelve más abrupto, es decir, hay películas que son descubiertas por muchos o incluso todos los usuarios, y películas que no son conocidas por prácticamente ninguno. Sin embargo, si sólo se habla con un amigo, el descubrimiento sigue una distribución prácticamente constante, esto es, todas las películas son descubiertas por un número similar de usuarios.

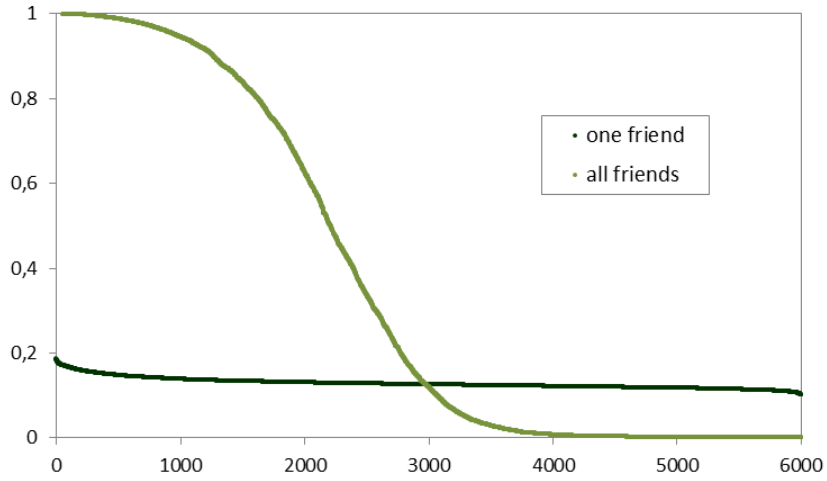


Figura 19. Curva de descubrimiento en función de si los usuarios hablan con todos sus amigos o sólo con uno cuando se comunican.

La explicación es sencilla, cuando un usuario decide comunicar a todos sus amigos la película que acaba de ver, esta película pasa a ser conocida, y por tanto, susceptible de ser vista y comunicada, por todos ellos. Cuantos más usuarios conozcan una película, más probable es que el resto de usuarios acaben conociéndola también, pues hay más usuarios que pueden hablar de ella. Además, según las películas se trasmiten de usuario en usuario, se van generando ratings; por tanto, cuanto mayor es la velocidad de

transmisión, antes se alcanza el millón de ratings (punto final de la simulación) y en ese momento hay películas que ni siquiera han sido descubiertas, de ahí el fuerte sesgo entre unas películas y otras.

Sin embargo, si el usuario sólo habla de la película con un amigo, el número de usuarios que la conocen se mantiene bajo y la probabilidad de ser descubierta por otros usuarios no se ve significativamente incrementada frente al resto de películas. La velocidad de generación de ratings también disminuye, dando tiempo a todas las películas a ser descubiertas mediante el descubrimiento exógeno. De ahí que la gráfica sea tan similar a una constante: todas las películas se descubren y no se producen fuertes sesgos en el nivel de descubrimiento de unas a otras.

También es interesante comparar cómo se relacionan la distribución del descubrimiento y la de la relevancia en ambos casos, es decir, si son las películas descubiertas las más relevantes o no. En la Figura 20 podemos observar dichas distribuciones de relevancia superpuestas sobre las de descubrimiento, para el caso de un amigo y de varios.

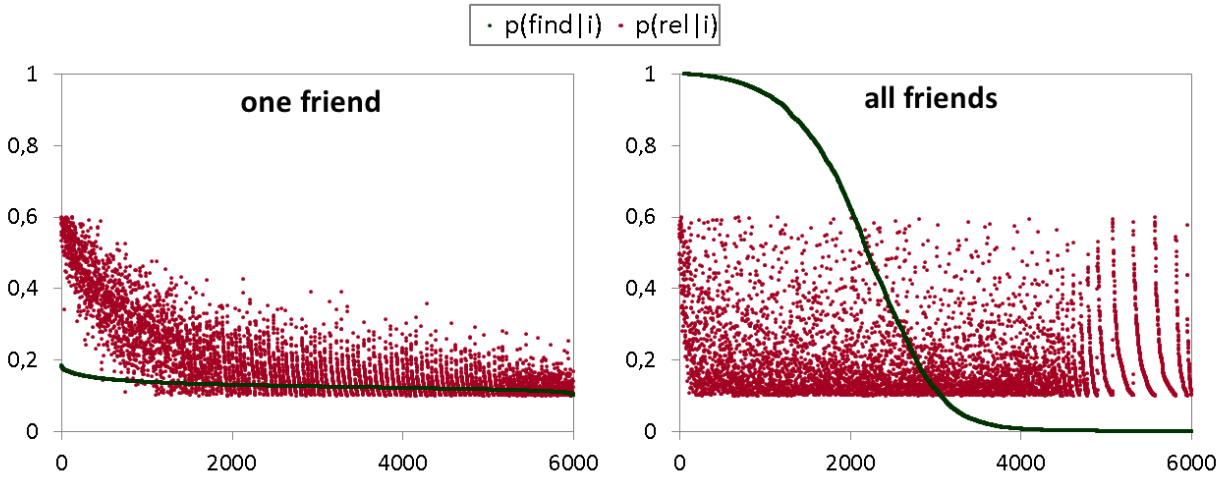


Figura 20. Curva de descubrimiento y distribución de relevancia en función de si los usuarios hablan con todos sus amigos o sólo con uno cuando se comunican.

Observamos que, cuando los usuarios se comunican con un único amigo, existe una correlación más alta entre relevancia y descubrimiento que cuando se comunican con todos sus vecinos. Aunque en las gráficas se aprecia esta correlación con bastante claridad, utilizamos el coeficiente de correlación de Pearson para comprobarlo analíticamente.

Dadas dos variables aleatorias, X e Y, el coeficiente de correlación de Pearson ρ_{XY} se calcula de la siguiente forma:

$$\rho_{XY} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \sigma_Y}$$

Donde σ_X y σ_Y son las desviaciones típicas de X e Y, μ_X y μ_Y las medias y σ_{XY} la covarianza entre X e Y. Se cumple que $\rho_{XY} \in [-1,1]$ y se interpreta de la siguiente forma: si $\rho_{XY} > 0$, existe una correlación positiva entre ambas variables, es decir, cuando una se incrementa la otra también. Cuanto mayor es ρ_{XY}, mayor es la correlación. Si $\rho_{XY} < 0$, existe una correlación negativa y ocurre el caso contrario, cuando una aumenta la otra disminuye.

En el caso de las variables relevancia y descubrimiento, el coeficiente de correlación de Pearson para la comunicación con un único amigo es 0,84 y para la comunicación con varios amigos es 0,17. Es decir, tal y como se esperaba, existe una correlación positiva mucho mayor en el caso de un solo amigo que en el de varios. Por tanto observamos que cuanto más activa es la comunicación en la red (los usuarios hablan frecuentemente con todos sus contactos), más influye la calidad de los ítems en su difusión. Y viceversa, en una red con poca comunicación (los usuarios hablan con sólo un contacto a la vez) los ítems tienden a difundirse independientemente de la satisfacción que en ellos encuentran los usuarios.

Intuitivamente, el motivo por el que ocurre esto es el siguiente. En el caso de un solo amigo, todas las películas acaban siendo descubiertas con probabilidad similar y, una vez que todas están en la red social, se transmiten más (y por tanto se descubren más) las más relevantes que las no relevantes porque $p(tell|R) > p(tell|\neg R)$. Es decir, los usuarios tienen más películas entre las que elegir cuál comunicar, en cuyo caso tienden a elegir más las que les resultan relevantes que las que no. Sin embargo, en el caso de varios amigos, una vez que se descubre una película, ésta se trasmite por la red social a gran velocidad, independientemente de su relevancia, porque son muy pocas las películas entre las cuales los usuarios pueden elegir

6.4.3 Nivel de descubrimiento exógeno

Probemos ahora a variar el nivel de descubrimiento exógeno, representado por el ratio entre películas descubiertas y vistas. En la Figura 21 observamos las curvas de descubrimiento que se obtienen para los valores 0.01, 0.1 y 1 de dicho ratio. Vemos que, cuanto mayor es el descubrimiento exógeno, más se asemejan las gráficas a una constante, esto es, las películas menos conocidas aumentan su descubrimiento y las más conocidas lo disminuyen.

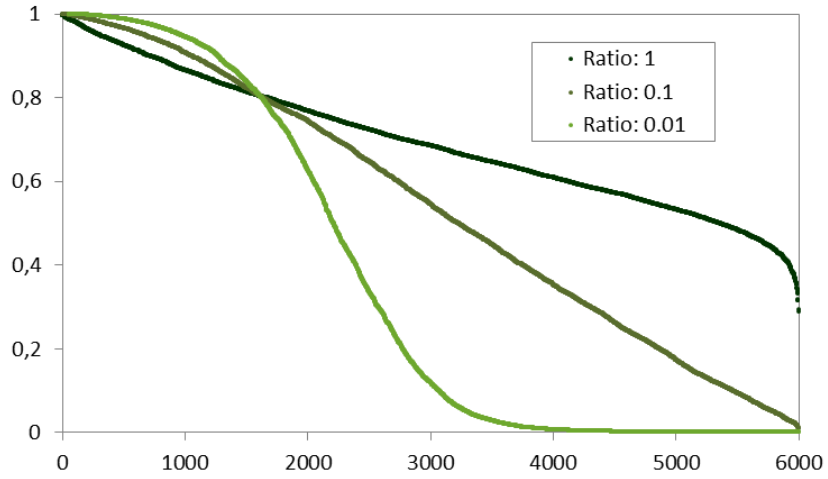


Figura 21. Curva de descubrimiento en función del ratio entre películas vistas y descubiertas.

El motivo es el siguiente: Con un nivel bajo de descubrimiento exógeno, la probabilidad de que una película que todavía no haya sido descubierta por nadie se descubra es muy baja, es decir, el ritmo al que se descubren nuevas películas es lento. Por ello, las pocas películas que sí han sido descubiertas, son las únicas que pueden ser trasmitidas por la red social, aumentando así rápidamente su nivel de descubrimiento frente al resto. Al aumentar el descubrimiento exógeno, todas las películas tienen más probabilidad de ser descubiertas, por lo que el número total de películas descubiertas

aumenta. Así, los usuarios tienen más posibilidades a la hora de elegir qué película consumir y transmitir y no se producen grandes sesgos de unas películas a otras.

Nuevamente, analizamos la distribución de relevancia en cada uno de los tres casos y su relación con el descubrimiento. El coeficiente de correlación de Pearson es 0, 17 (ratio 0.01), 0,42 (ratio 0.1) y 0,68 (ratio 1) respectivamente. Es decir, tal y como se observa en las gráficas de la Figura 22, cuanta más información externa fluye hacia la red, mayor es la correlación entre relevancia y descubrimiento.

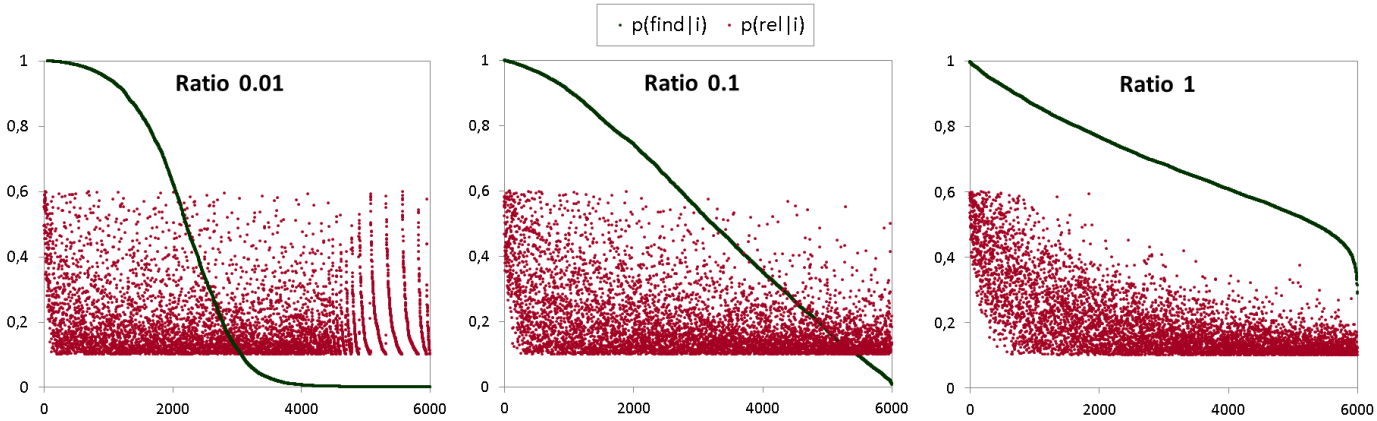


Figura 22. Curva de descubrimiento y distribución de relevancia en función del ratio entre películas vistas y descubiertas.

El motivo es el mismo al que ocurría para el caso un único amigo vs. varios amigos. Cuanto menor es el ratio de descubrimiento, los usuarios limitan las películas que transmiten a las que les han trasmitido otros usuarios. El número es muy limitado, y la relevancia apenas interviene, es decir, se transmiten las pocas películas que se conocen, independientemente de si gustan o no.

6.4.4 Grafo

Hemos observado en las pruebas empíricas que las diferencias entre los grafos Barabási y Erdös prácticamente son inapreciables, por lo que, a la hora de estudiar la influencia del grafo en los resultados de la simulación únicamente se ha comparado el grafo Barabási con un grafo de Facebook[17] de 3.959 usuarios. Hemos generado además dos grafos adicionales partiendo del de Facebook, eliminando usuarios de forma manual con el objetivo de generar grafos más pequeños de naturaleza similar. El motivo por el que estudiamos estos dos grafos más pequeños es para tener una impresión informal de cómo influye el tamaño en los resultados. Los datos de los grafos utilizados se pueden observar en la Tabla 11, junto con el de Barabási.

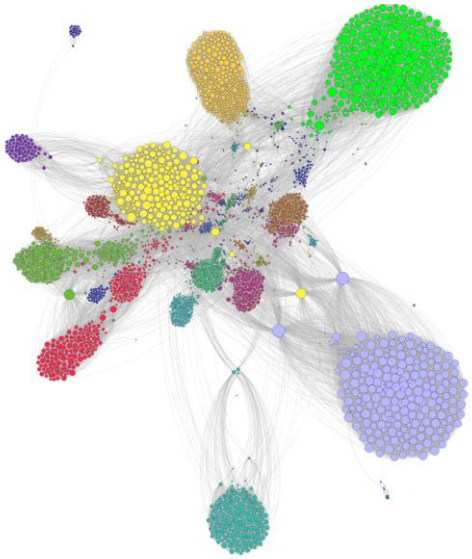
Grafo	Nodos	Aristas	Coeficiente de clustering
Barabási	4.000	159.600	0,056
Facebook	3.956	170.174	0,557
Facebook	1.665	66.976	0,549
Facebook	648	6.855	0,576

Tabla 11 Características de los grafos Barabási y Facebook, incluyendo los subgrafos de este último.

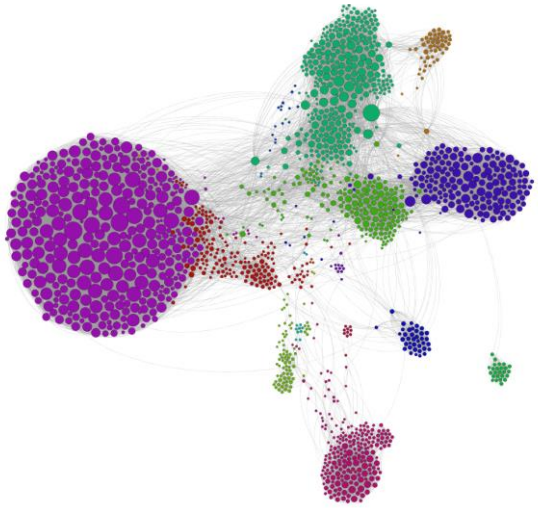
[17] http://snap.stanford.edu/data/egonets-Facebook.html

En la Figura 23 se muestra la topología del grafo y los subgrafos de Facebook. La disposición geométrica (layout) se ha generado con el algoritmo Force Atlas 2 para visualización disponible en la herramienta Gephi. Los colores de los nodos representan clusters generados con el algoritmo "modularity" disponible en esta misma herramienta, que identifica clusters de densidad de conexiones.

a) Grafo de Facebook con 3959 usuarios



b) Subgrafo del grafo a) con 1665 usuarios



c) Subgrafo del grafo a) con 648 usuarios

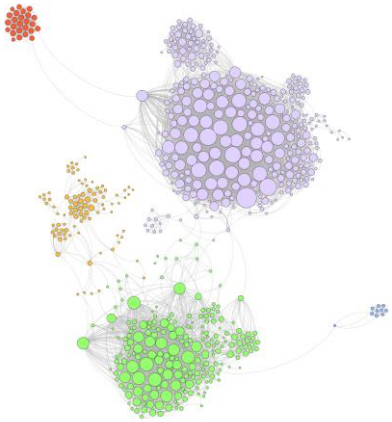


Figura 23. Grafo y subgrafos de Facebook.

En la Figura 24 mostramos las curvas de descubrimiento que se producen ejecutando la simulación en los grafos de Barabasi y Facebook. Como el número de usuarios es distinto en cada grafo, se ha ajustado linealmente el número de ratings en los que parar la simulación, para que el comportamiento sea comparable. Es decir, se está asumiendo el ratio 1000000 ratings / 4000 usuarios para calcular el momento de parada de los grafos con menos usuarios. Así, para el grafo de Facebook de 1665 usuarios, se ha detenido la simulación a los 400000 ratings, y para el de 648 usuarios se ha detenido en 160000 ratings.

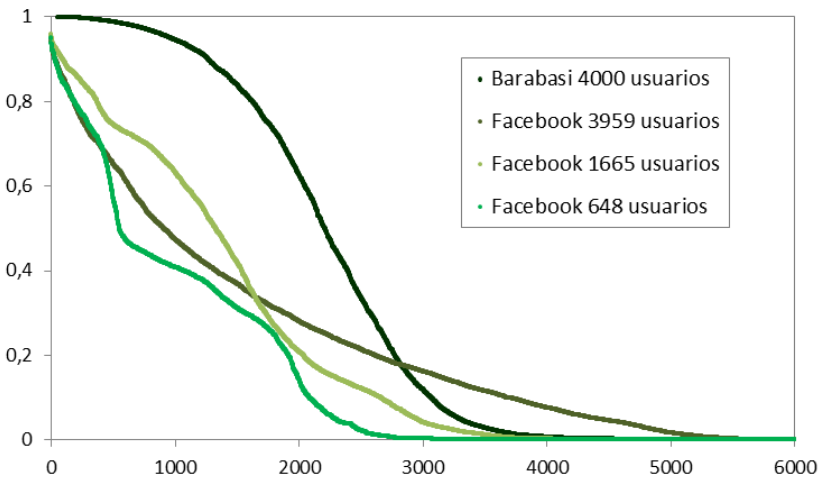


Figura 24. Comparativa de las curvas de descubrimiento en función del tipo de grafo.

Observamos grandes diferencias entre las curvas. Las gráficas de los grafos de Facebook presentan un comportamiento más lineal que la de Barabási. Además, en los subgrafos de Facebook se producen abultamientos en la gráfica. Intuitivamente y observando en la Figura 23 los grafos obtenidos al eliminar usuarios, parece que cada elevación se corresponde con uno de los grandes clusters. Así, en el grafo de 1665 usuarios hay un cluster que destaca por su tamaño frente al resto, y en la curva de descubrimiento hay una elevación más fuerte que las demás. En el grafo de 648 usuarios, por su parte, destacan dos clusters principalmente, y en la curva se observan dos elevaciones. En el grafo de Facebook de 4000 usuarios, sin embargo, hay varios clusters con tamaños similares, lo que podría justificar la ausencia de elevaciones perceptibles.

Una posible explicación de este comportamiento sería la siguiente: cuando una película es descubierta por un usuario de una componente, dicho usuario transmite la información a sus vecinos y al final, prácticamente todos los usuarios del cluster descubren rápidamente la película. De esta forma, todas las películas que caigan en dicho cluster sumaran un valor similar a su descubrimiento, produciéndose así las distintas elevaciones que observamos en la curva.

Respecto al motivo por el cual la curva de Barabasi es mucho más escarpada que la del resto de grafos, parece provenir nuevamente de la ausencia de clusters en comparación con los grafos de Facebook. Así, tal y como se observa en la Taba 11, el coeficiente de clustering de Barabasi es un orden menor que el del resto de grafos.

Sin embargo, confirmar estas ideas intuitivas sería objeto de estudio de un posible trabajo futuro y más allá del alcance de este trabajo.

7. Conclusiones

El presente trabajo se desarrolla tomando como objetivo la investigación del efecto que las dinámicas de comunicación en redes sociales puede tener en la generación y distribución de datos para la ejecución y evaluación de sistemas de recomendación. El trabajo desarrollado abarca aspectos de formulación y análisis de modelos, la configuración de procesos de simulación, y la implementación de software para la ejecución de simulaciones sobre estructuras de red, la visualización dinámica interactiva de las mismas mediante interfaces de usuario a medida, y el análisis estadístico de resultados y el ajuste de datos reales. Recapitulamos a continuación los aspectos más relevantes del trabajo realizado, y planteamos las múltiples líneas de trabajo futuro que se abren a partir del trabajo desarrollado hasta aquí.

7.1 Resumen y contribuciones

Los datos que se utilizan en la ejecución y evaluación de algoritmos de recomendación tienen fuertes sesgos en la distribución de las observaciones. Es relevante, por tanto, entender cómo se generan estos sesgos de cara a ajustar y evaluar dichos algoritmos. El trabajo documentado en esta memoria ha consistido en diseñar un escenario, y un conjunto de modelos y herramientas para materializarlo, que permita estudiar dichas distribuciones y en particular la influencia que tienen sobre ellas los fenómenos de propagación de información en redes sociales.

El escenario desarrollado es una generalización de otros modelos propuestos y estudiados por otros autores (Newman 2010, Doerr 2012) los cuales han sido utilizados para contrastar la verosimilitud del modelo aquí planteado. Entre las diferencias originales del modelo que desarrollamos, destacamos la inclusión de los gustos de los usuarios, lo cual permite comparar la distribución resultante de votos observados con las opiniones reales (observadas o no) de dichos usuarios.

Una vez diseñado, el modelo ha sido implementado en una aplicación Java haciendo uso de diversas y variadas librerías de dominio público. Este programa permite visualizar las distribuciones que se obtienen en tiempo real mediante una simulación. Adicionalmente, se ha desarrollado una implementación auxiliar en Javascript que permite una visualización cualitativa más detallada de la red social y de los diversos estados que atraviesan los usuarios durante la simulación.

Una vez implementado, y a fin de validar la verosimilitud y potencial explicativo de nuestra propuesta, se ha utilizado el modelo para ajustar distribuciones de conjuntos de datos públicos del campo de los sistemas de recomendación (MovieLens, Twitter, Epinions y Foursquare) con el objetivo de recrear el posible proceso y la combinación de factores que se esconden tras las distribuciones resultantes que se observan en estas colecciones.

Por último, se ha realizado un análisis preliminar de las relaciones y dependencias entre los parámetros del modelo y las variables de salida resultantes. En particular, se ha visto que la comunicación entre los usuarios influye muy notablemente en cómo se distribuye el descubrimiento de información y, con él, la generación de ratings.

También la forma de la red social parece ser clave en la distribución del descubrimiento y en particular en la velocidad a la que se produce.

Sin embargo, aún queda un amplio espectro de parámetros por explorar, e hipótesis que desarrollar, lo que hace que este trabajo sirva como base para multitud de trabajos futuros ya que inicia una senda apenas explorada hasta el momento.

7.2 Trabajo futuro

A lo largo del trabajo se han ido citando varias posibilidades y variables de interés para un posible trabajo futuro. En este apartado se exponen de forma sintetizada todas ellas y algunas más que no se han mencionado explícitamente hasta ahora. Se han clasificado en dos grupos, según atiendan a mejorar el modelo o el análisis de los resultados.

7.2.1 Modelo

En relación al modelo planteado existen muchas variables que se podrían añadir con el objetivo de aumentar su complejidad y por tanto su semejanza con la realidad. Enumeramos a continuación las que consideramos más relevantes:

- Evolución y formación de preferencias. Una de las asunciones de nuestro modelo es que la relevancia, esto es, los gustos de los usuarios, permanece constante a lo largo de todo el proceso. Sin embargo, en la realidad nuestros gustos evolucionan en función de diversas influencias, como los gustos de nuestros amigos, la publicidad, las críticas, etc. Por tanto, un posible trabajo futuro sería introducir en el modelo una relevancia variable o diseñar otro escenario distinto que permita estudiar la evolución y formación de dicha relevancia en función de distintos parámetros.
- Introducción de nuevos ítems. Otra posible mejora del modelo sería incorporar la introducción de nuevos ítems al sistema, simulando así el estreno de nuevas películas. Esto podría introducir un sesgo hacia los ítems que lleven más tiempo en el sistema, pues tienen más oportunidades de ser descubiertos. Una posible solución consistiría en hacer depender la probabilidad de descubrir un ítem del tiempo que lleve en el sistema, de forma que cuanto más nuevo sea más probable es que se descubra. Esto concuerda con lo que ocurre en la realidad, donde por regla general las nuevas películas tienen más probabilidad de ser vistas en el presente, aunque en total han sido menos vistas que las viejas puesto que han tenido menos tiempo de existencia.
- Actividad de los usuarios no uniforme. En el modelo diseñado todos los usuarios tienen la misma probabilidad de intervenir en las diferentes acciones que se contemplan, sin embargo, podría considerarse una actividad sesgada en la que, como suele suceder en la realidad, unos usuarios son más activos que otros. Por ejemplo, podríamos considerar que los usuarios con más vecinos tiendan a intervenir más, dado que tienen más influencia y por tanto la distribución final debería depender más de su actividad.
- Decisión bajo influencia. En el modelo formulado, las decisiones de los usuarios son independientes de sus fuentes de información. Sin embargo en la realidad tendemos a prestar diferente atención a las opciones que se nos presentan según su procedencia. Por ejemplo, suele ser más frecuente que decidamos ir a ver una película si nos la recomienda un amigo que si nos la recomienda un spot de publicidad. Diferentes personas a su vez pueden tener diferente grado de influencia sobre sus amigos.

- Redes de estructura dinámica. Las redes reales no permanecen estáticas mientras transcurren los procesos a través de ellas, sino que constantemente se crean o refuerzan nuevos enlaces y desaparecen otros, aparecen nuevos usuarios, comunidades, etc. Los procesos de comunicación afectan de hecho retroactivamente en la formación evolución de las estructuras de red.

7.2.2 Análisis de resultados

Con respecto a la exploración de parámetros, en este trabajo se ha realizado un primer acercamiento, pero sería relevante realizar un análisis más detallado de todos los parámetros para estudiar su papel en los resultados. En particular, queda abierta la hipótesis de cómo influye la forma de la red social y su coeficiente de clustering sobre la curva de descubrimiento.

Otro tema que abre muchas y nuevas posibilidades atañe a la evaluación y ejecución de recomendadores en la que podemos distinguir dos objetivos principalmente.

- Evaluación: El modelo permite no sólo incorporar nuevos algoritmos de recomendación sino también evaluarlos con distintas métricas, las cuales también se pueden configurar. Esto permite utilizar la aplicación como herramienta de evaluación de recomendadores.
- Análisis de la influencia: Otra posibilidad es considerar la influencia de recomendadores en el sistema, no sólo su evaluación. Así, se podría analizar cómo varían los resultados en función de los distintos recomendadores y sus parámetros y estudiar el posible efecto recursivo de algunos de ellos: los recomendadores recomienden las películas más populares, lo cual aumenta su popularidad y hace que se recomienden más.

En resumen, el modelo ofrece amplias posibilidades en el ámbito de la recomendación tanto para evaluar algoritmos como para analizar su influencia y los sesgos que pueden producir en las distribuciones de ratings.

Referencias

J. Scott, P. J. Carrington (Eds). The SAGE Handbook of Social Network Analysis, 1st Edition, SAGE , 2011.

H. Steck. Training and testing of recommender systems on data missing not at random. 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2010). Washington, DC, USA, July 2010, pp. 713-722

P. Cremonesi, Y. Koren, R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. Fourth ACM Conference on Recommender Systems (RecSys 2010). Barcelona, Spain, September 2010 pp. 39-46

M. Blattner and M. Medo. Recommendation Systems in the Scope of Opinion Formation: a Model. CEUR Workshop Vol-893, Decisions@RecSys, 2nd Workshop on Human Decision Making in Recommender Systems in conjunction with the 6th ACM Conference on Recommender Systems (RecSys 2012), Dublin, Ireland, September 2012, pp. 32-39

B. Doerr, M. Fouz, T. Friedrich. Why Rumors Spread So Quickly in Social Networks. Communications of the ACM 55(6), June 2012, pp. 70-75.

A. Anagnostopoulos, R. Kumar, M. Mahdian. Influence and Correlation in Social Networks. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008). Las Vegas, NV, USA, August 2008, pp. 7-15

P. N. Howard, A. Duffy, D. Freelon, M. Hussain, W. Mari, M. Mazaid. Opening Closed Regimes: What Was the Role of Social Media During the Arab Spring?. Project on Information Technology and Political Islam, Washington, DC, USA, September 2011, pp 30

P. Erdös and A. Rényi. On Random Graphs I. Publicationes Mathematicae 6, 1959, pp. 290–297.

A.-L. Barabási and R. Albert. Emergence of scaling in random networks, Science 286 (5439), October 1999, pp. 509-512.

M.E.J. Newman. Networks, An Introduction, 1st Edition. Oxford University Press, 2010

G. Adomavicius and Y. Kwon. Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques. IEEE Transactions on Knowl.edge and Data Engineering 24(5), May 2012, pp. 896-911.

F. Ricci, L. Rokach, L., B. Shapira, P. B. Kantor (Eds.). Recommender Systems Handbook, 1st Edition. Springer, 2011

H. Steck. Item popularity and recommendation accuracy. 5th ACM Conference on Recommender Systems (RecSys 2011). Chicago, IL, October 2011, pp. 125-13

T. B. Arnold and J.W. Emerson. Nonparametric Goodness-of-Fit Tests for Discrete Null Distributions. R Journal 3(2), December 2011, pp. 34-39.

P. E. Greenwood, M. S. Nikulin. A Guide to Chi-Squared Testing. John Wiley & Sons, 1996.

A. Alhambra. Recomendación de contactos en Twitter. Trabajo Fin de Grado, Escuela Politécnica Superior, Universidad Autónoma de Madrid, junio 2014.